Chapitre 1

Complexity and approximation results for bounded-size paths packing problems

1.1. Introduction

This chapter presents some recent works given by the authors ([MON 07a, MON 07b]) about the complexity and the approximation of several problems on computing collections of (vertex)-disjoint paths of bounded size.

1.1.1. *Bounded-size paths packing problems*

A $\mathbf{P}_k$ partition of the vertex set of a simple graph $G = (V, E)$ is a partition of V into q subsets $V_1, \cdots, V_q$, each of size $|V_i| = k$, such that the subgraph $G[V_i]$ induced by any V_i contains a Hamiltonian path. In other words, the partition $(V_1, \ldots, V_q)$ describes a collection of $|V|/k$ vertex disjoint simple paths of length $k-1$ (or, equivalently, simple paths on k vertices) on G. The decision problem called $\mathbf{P}_k$ partitioning problem ($\mathbf{P}_k$PARTITION in short) consists, given a simple graph $G = (V, E)$ on $k \times n$ vertices, in deciding whether G admits or not such a partition. The analogous problem where the subgraph $G[V_i]$ induced by V_i is isomorphic to $\mathbf{P}_k$ (the chordless path on k vertices) will be denoted by INDUCED $\mathbf{P}_k$PARTITION. These two problems are **NP**-complete for any $k \geq 3$, and polynomial otherwise, [GAR 79, KIR 78]. In fact, they both are a particular case of a more general problem called *partition into isomorphic subgraphs*, [GAR 79]. In [KIR 78], Kirkpatrick and Hell give a necessary and sufficient condition for the **NP**-completeness of the partition into isomorphic

Chapitre rédigé par Jérôme MONNOT and Sophie TOULOUSE.

subgraphs problem in general graphs. $\mathbf{P}_k$PARTITION has been widely studied in the literature, mainly because of its closeness to two famous optimization problems, namely : the minimum k-path partition problem (denoted by MINk-PATHPARTITION) and the maximum $\mathbf{P}_k$ packing problem (denoted by MAX$\mathbf{P}_k$PACKING).

On the one hand, MINk-PATHPARTITION can be viewed as an optimization version of $\mathbf{P}_k$PARTITION where the constrainst on the exact length of the paths is relaxed. MINk-PATHPARTITION consists in partitioning the vertex set of a graph $G = (V, E)$ into the smallest number of paths so that each path has *at most* k vertices (for instance, MIN2-PATHPARTITION is equivalent to the maximum matching problem). The optimal value is usually denoted by $\rho_{k-1}(G)$ for any $k \geq 2$, by $\rho(G)$ when no constraint occurs on the length of the paths (in particular, $\rho(G) = 1$ *iff* G has a Hamiltonian path). MINk-PATHPARTITION has been extensively studied in the literature, [STE 03, STE 00, YAN 97], and has applications in broadcasting problems (see for example [YAN 97]).

On the other hand, if we relax the exact covering constraint, then we obtain the optimization problems MAX$\mathbf{P}_k$PACKING and MAXINDUCED$\mathbf{P}_k$PACKING which consist, given a simple graph $G = (V, E)$, in finding a maximum number of vertex-disjoint (induced) $\mathbf{P}_k$. When considering the weighted case (denoted by MAXW$\mathbf{P}_k$PACKING and MAXWINDUCED$\mathbf{P}_k$PACKING, respectively), the input graph $G = (V, E)$ is given together with a weight function w on its edges, and the goal is to find a collection $\mathcal{P} = \{P_1, \ldots, P_q\}$ of vertex-disjoint (induced) $\mathbf{P}_k$ that maximizes $w(\mathcal{P}) = \sum_{i=1}^{q} \sum_{e \in P_i} w(e)$.

The special case of MAXW$\mathbf{P}_k$PACKING where the graph is complete on $k \times n$ vertices is called the weighted $\mathbf{P}_k$ partition problem ($\mathbf{P}_k$P in short). In this case, each solution contains exactly n vertex disjoints paths of length $k-1$. If the goal is to maximize (MAX$\mathbf{P}_k$P), then we seek a $\mathbf{P}_k$ partition of maximum weight, and if the goal is to minimize (MIN$\mathbf{P}_k$P), then we seek a $\mathbf{P}_k$ partition of minimum weight. When considering the minimization version, it is more often assumed that the instance is metric, i.e., that the weight function satisfies the triangle inequality : $w(x, y) \leq w(x, z) + w(z, y), \forall x, y, z$; MINMETRIC$\mathbf{P}_k$P will refer to this restriction. Note that this latter version of the problem is closely related to the vehicle routing problem when restricting the route of each vehicle to at most k intermediate stops, [ARK 06, FRE 78]. Finally, we also will consider the special case of metric instances where the weight function is either 1 or 2 ; the corresponding problems will be denoted by MAX$\mathbf{P}_k$P$_{1,2}$ and MIN$\mathbf{P}_k$P$_{1,2}$ ($\mathbf{P}_k$P$_{1,2}$ when the goal is not specified). Such a restriction makes sense, since it provides an alternative relaxation of the initial decision problem $\mathbf{P}_k$Partition ; moreover, MIN$\mathbf{P}_k$P$_{1,2}$ and MINk-PATHPARTITION are strongly connected.

All theses problems are very closed one to each other. In particular, $\mathbf{P}_k$PARTITION **NP**-completeness implies the **NP**-hardness of both MINk-PATHPARTITION and $\mathbf{P}_k$P (even when restricting to $\mathbf{P}_k\mathrm{P}_{1,2}$) ; conversely, $\mathbf{P}_k$PARTITION is polynomial-time decidable on instance families where MINk-PATHPARTITION or MAX$\mathbf{P}_k$PACKING are polynomial-time computable.

1.1.2. *Complexity and approximability status*

The minimum k-path partition problem is obviously **NP**-complete in general graphs [GAR 79], and remains intractable in comparability graphs, [STE 03], in cographs, [STE 00], and in bipartite chordal graphs, [STE 03] (when k is part of the input). Note that most of the proofs of **NP**-completeness actually establish the **NP**-completeness of $\mathbf{P}_k$PARTITION. Nevertheless, the problem turns out to be polynomial-time solvable in trees, [YAN 97], in cographs when k is fixed, [STE 00] and in bipartite permutation graphs, [STE 03]. Note that one can also find in the literature several results about the problem that consists in partitioning the graph into disjoints paths of length at least 2, [WAN 94, KAN 03].

This chapter proposes new complexity and inapproximability results for (INDUCED) $\mathbf{P}_k$PARTITION, MINk-PATHPARTITION and MAX(W)(INDUCED)$\mathbf{P}_k$PACKING, mostly in the case of bipartite graphs, discussing the graph maximum degree. Namely, we study the case of bipartite graphs of maximum degree 3 : first, these problems are **NP**-complete for any $k \geq 3$ (and this even if the graph is planar, for $k = 3$); second, there is no **PTAS** for MAX(INDUCED)$\mathbf{P}_k$PACKING or, more precisely, there is a constant $\varepsilon_k > 0$ such that it is **NP**-hard to decide whether a maximum (induced) $\mathbf{P}_k$-packing is of size n or of size upper bounded by $(1 - \varepsilon_k)n$. On the opposite side, all these problems trivially become polynomial-time computable both in graphs of maximum degree 2 and in forests.

Where these problems are intractable, what about their approximation level ? We recall that a given problem is said to be ε-approximable if it admits an algorithm that polynomially computes on any instance a solution that is at least (if maximizing, at most if minimizing) ε times the optimum value. To our knowledge, there is no specific approximation result for neither MINk-PATHPARTITION, nor MAXW$\mathbf{P}_k$PACKING, in general graphs. Nevertheless, one can find some approximation results for the k-path partition problem where the objective consists in maximizing the number of edges of the paths that participate to the solution (see [VIS 92] for the general case, [CSA 02] for dense graphs). Concerning MAXW$\mathbf{P}_k$PACKING, using approximation results for the maximum weighted k-packing problem (mainly based on local search techniques), [ARK 98], one can obtain a $(\frac{1}{k-1} - \varepsilon)$-approximation; in particular, MAXW$\mathbf{P}_3$PACKING is $(\frac{1}{2} - \varepsilon)$-approximable.

In the case of complete graphs, MAX$\mathbf{P}_k$P is standard-approximable for any k, [HAS 97]. In particular, MAX$\mathbf{P}_3$P and MAX$\mathbf{P}_4$P are respectively $35/67 - \varepsilon$, [HAS 06] and $3/4$, [HAS 97] approximable. Note that for $k = 2$, a P_2-partition is a perfect matching and hence, MINP$_2$P and MAXP$_2$P both are polynomial-time computable. The minimum case is trickier : from the fact that $\mathbf{P}_k$PARTITION is **NP**-complete in general graphs, it is **NP**-hard to approximate MIN$\mathbf{P}_k$P within $2^{p(n)}$ for any polynomial p, for any $k \geq 3$. Nevertheless, one could expect that the metric instances are constant-approximable, even though no approximation rate (to our knowledge) has been established so far for MINMETRIC$\mathbf{P}_k$P.

Here, we provide new approximation results for MIN3-PATHPARTITION, MAXW-$\mathbf{P}_3$PACKING and $\mathbf{P}_k$P. Concerning the two former problems, we propose a $3/2$-approximation for MIN3-PATHPARTITION in general graphs and a $1/3$ (*resp.*, a $1/2$)-approximation for MAXW$\mathbf{P}_3$PACKING in general (*resp.*, bipartite) graphs of maximum degree 3. But we more focus on $\mathbf{P}_k$P, and more specifically on $\mathbf{P}_4$P, by analyzing the performance of a specific algorithm proposed by Hassin and Rubinstein, [HAS 97], under different assumptions on the input. Doing so, we put to the fore the effectiveness of this algorithm by proving that it provides new approximation ratios for both standard and differential measures, for both maximization and minimization versions of the problem. But, before going so far, we briefly recall the basis of approximation theory, introduce some notations and then give this outline of the chapter.

1.1.3. *Theoritical framework, notations and organization*

Consider an instance I of an **NP**-hard optimization problem Π and a polynomial-time algorithm $\mathtt{A}$ that computes feasible solutions for Π. Denote by $\mathrm{apx}_\Pi(I)$ the value of a solution computed by $\mathtt{A}$ on I, by $\mathrm{opt}_\Pi(I)$ the value of an optimal solution and by $\mathrm{wor}_\Pi(I)$ the value of a worst solution (that corresponds to the optimum value when reversing the optimization goal). The quality of $\mathtt{A}$ is expressed by means of approximation ratios that somehow compare the approximate value to the optimum one. So far, two measures stand out from the literature : the *standard* ratio [AUS 99] (the most widely used) and the *differential* ratio [AUS 80, BEL 95, DEM 96, HAS 01]. The standard ratio is defined by $\rho_\Pi(I, \mathtt{A}) = \mathrm{apx}_\Pi(I)/\mathrm{opt}_\Pi(I)$ if Π is a maximization problem, by $\rho_\Pi(I, \mathtt{A}) = \mathrm{opt}_\Pi(I)/\mathrm{apx}_\Pi(I)$ otherwise, whereas the differential ratio is defined by $\delta_\Pi(I, \mathtt{A})$= $(\mathrm{wor}_\Pi(I) - \mathrm{apx}_\Pi(I))/(\mathrm{wor}_\Pi(I) - \mathrm{opt}_\Pi(I))$. In other words, the standard ratio divides the approximate value by the optimum one, whereas the differential ratio divides the distance from a worst solution to the approximate value by the instance diameter.

Within the worst case analysis framework and given a universal constant $\varepsilon \leq 1$ (*resp.*, $\varepsilon \geq 1$), an algorithm $\mathtt{A}$ is said to be an ε-standard approximation for a maximization (*resp.* a minimization) problem Π if $\rho_{I,\mathtt{A}_\Pi}(I) \geq \varepsilon \ \forall I$ (*resp.*, $\rho_{\mathtt{A}_\Pi}(I) \leq \varepsilon$ $\forall I$). With respect to differential approximation, $\mathtt{A}$ is said to be ε-differential approximate for Π if $\delta_{\mathtt{A}_\Pi}(I) \geq \varepsilon, \ \forall I$, for a universal constant $\varepsilon \leq 1$. Equivalently, seing any solution value as a convex combination of the two values $\mathrm{wor}_\Pi(I)$ and $\mathrm{opt}_\Pi(I)$, an approximate solution value $\mathrm{apx}_\Pi(I)$ will be ε-differential approximate if for any instance I, $\text{apx}_\Pi(I) \geq \varepsilon \times \text{opt}_\Pi(I) + (1-\varepsilon) \times \text{wor}_\Pi(I)$ (for the maximization case ; reverse the sense of the inequality when minimizing). For both measures, a given problem Π is said to be constant approximable if there exists a polynomial-time algorithm $\mathtt{A}$ and a universal constant ε such that $\mathtt{A}$ is an ε- approximation for Π. The class of problems that are standard- (*resp.*, differential-) constant-approximable is denoted by **APX** (*resp.*, by **DAPX**). If Π admits a polynomial-time approximation scheme, that is, a whole algorithm family $(\mathtt{A}_\varepsilon)_{(\varepsilon)}$ such that $\mathtt{A}_\varepsilon$ is ε-approximate for any ε (note that the time-complexity of $\mathtt{A}_\varepsilon$ may be exponential in $1/|1-\varepsilon|$), then Π belongs to the class **PTAS** (*resp.*, **DPTAS**).

The notations that will be used are the usual ones according to graph theory. Moreover, we exclusively work in undirected simple graphs. In this chapter, we often identify a path P of length $k-1$ with $\mathbf{P}_k$, even if P contains a chord. However, when dealing with INDUCED $\mathbf{P}_k$PARTITION, the paths that are considered are chordless. Finally, when no ambiguity occurs on the problem that is concerned, we will omit the reference to Π to denote the values $\text{apx}(I)$, $\text{opt}(I)$ and $\text{wor}(I)$. For a better understanding of what follows, we recall some basic concepts of graph theory : a simple graph $G = (V, E)$ is said to be bipartite (or, equivalently, 2-colorable) if there exists a partition L, R of its vertex set such that E is contained in $L \times R$. A graph is planar if it can be drawn in the plane so that no edges intersect. A path (*resp.*, a cycle) $\Gamma = \{v_{j_1}, \ldots, v_{j_q}\} \subseteq E$ in G of length at least 2 (*resp.*, of length at least 4) is chordless if there is in E no other edge than the ones of Γ linking two vertices of Γ. G is chordal if none of its cycle of length at least 4 is chordless. G is an interval graph if one can associate to each vertex $v_j \in V$ an interval $[a_j, b_j]$ on the real line such that two intervals $[a_j, b_j]$ and $[a_\ell, b_\ell]$ intersect *iif* the edge $[v_j, v_\ell]$ belongs to E ; note that interval graphs are special cases of chordal graphs.

This chapter is organized as follows : the two next sections are dedicated to the study of (INDUCED) $\mathbf{P}_k$PARTITION, MAX(INDUCED)$\mathbf{P}_k$PACKING and MINk-PATHPARTITION. Section 1.2 focus on the complexity status of those problems in bipartite graphs, whereas Section 1.3 proposes some approximation results for MAXW$\mathbf{P}_3$PACKING and MIN3-PATHPARTITION. The fourth section is then dedicated to both standard and differential approximation of $\mathbf{P}_k$P. Subsection 1.4.1 provides a differential approximation for $\mathbf{P}_k$P while bridging some gap between differential approximation of TSP and differential approximation of $\mathbf{P}_k$P. Finally, Subsection 1.4.2, which constitutes the main part of Section 1.4, leads a complete analysis of the approximation level of an algorithm proposed by Hassin and Rubinstein [HAS 97], depending on the approximation measure that is considered and the characteristics of the input weight function.

The two main points of the chapter are, on the one hand, the establishment of new complexity results concerning $\mathbf{P}_k$PARTITION and related problems in bipartite graphs by means of reductions (section 1.2) and, on the other hand, the way the algorithm that is addressed in section 1.4.2 appears to be robust, in the sense that this latter provides

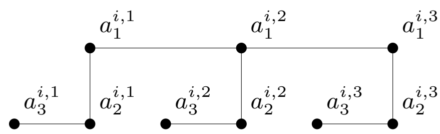


Figure 1.1. *The gadget $H(c_i)$ when c_i is a 3-tuple.*

good quality solutions (the best known so far), whatever version of the problem we deal with, whatever approximation framework within which we estimate the approximate solutions.

1.2. Complexity of $\mathbf{P}_k$PARTITION and related problems in bipartite graphs

1.2.1. *Negative results from the k-dimensional matching problem*

1.2.1.1. *k-dimensional matching problem*

The negative results we present all are based on a transformation from the k-dimensional matching problem, kDM, which is known to be **NP**-complete, [GAR 79]. An instance of kDM consists of a subset $\mathcal{C} = \{c_1, \ldots, c_m\} \subseteq X_1 \times \ldots \times X_k$ of k-tuples, where $X_1, \ldots, X_k$ are k pairwise disjoint sets of size n. A matching is a subset $M \subseteq \mathcal{C}$ such that no two elements in M agree in any coordinate, and the purpose of kDM is to answer the question : does there exist a perfect matching M on $\mathcal{C}$, that is, a matching of size n ? In its optimization version, the maximum k-dimensional matching problem (MAXkDM) addresses the question of computing a matching that is of maximal size.

1.2.1.2. *Transforming an instance of* kDM *into an instance of* $\mathbf{P}_k$PACKING

Let $I = (X_1, \ldots, X_k; \mathcal{C})$ be an instance of kDM, where $|X_q| = n,\ \forall q$ and $|\mathcal{C}| = m$. We denote by X the union of the element sets $X_1, \ldots, X_k$. Furthermore, for each element $e_j \in X$, we denote by d^j its degree, where the degree of an element e_j is defined as the number of k-tuples $c_i \in \mathcal{C}$ that contain e_j. We build an instance $G = (V, E)$ of INDUCED $\mathbf{P}_k$PACKING, where G is a bipartite graph of maximum degree 3, by associating a k-tuple gadget $H(c_i)$ to each k-tuple $c_i \in \mathcal{C}$, an element gadget $H(e_j)$ to each element $e_j \in X$, and then by linking the two gadget families by some edges. Our construction (more precisely, the element gadgets) depends on the parity of k.

1) *The element gadget $H(c_i)$*. For any k-tuple $c_i \in \mathcal{C}$, the gadget $H(c_i)$ consists of a collection $\left\{P^{i,1}, \ldots, P^{i,k}\right\}$ of k vertex-disjoint $\mathbf{P}_k$ with $P^{i,q} = \left\{a_1^{i,q}, \ldots, a_k^{i,q}\right\}$ for $q = 1, \ldots, k$, plus the edges $[a_1^{i,q}, a_1^{i,q+1}]$ for $q = 1$ to $k-1$. Hence, $H(c_i)$ contains

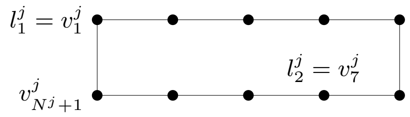


Figure 1.2. *The gadget $H(e_j)$ for $k = 3$ and $d^j = 2$.*

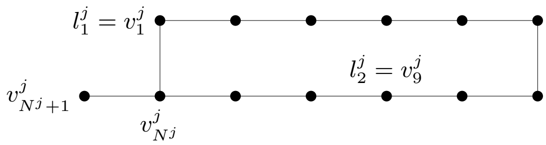


Figure 1.3. *The gadget $H(e_j)$ for $k = 4$ and $d^j = 2$.*

the k initial paths $P^{i,1}, \dots, P^{i,k}$, plus the additional path $\left\{a_1^{i,1}, \dots, a_1^{i,k}\right\}$. Figure 1.1 proposes an illustration of the k-tuple gadget when $k = 3$.

2) *The element gadget $H(e_j)$.* Let $e_j \in X$ be an element, with degree d^j. We distinguish two cases according to the parity of k.

– Odd values of k. $H(e_j)$ is defined as a cycle $\left\{v_1^j, \dots, v_{N^j+1}^j, v_1^j\right\}$ on $N^j + 1$ vertices, where $N^j = k(2d^j - 1)$. Moreover, for $p = 1$ to d^j, we denote by l_p^j the vertex of index $2k(p-1) + 1$. Thus, the element gadget is a cycle on a number of vertices that is a multiple of k plus 1, with d^j remarkable vertices l_p^j that will be linked to the k-tuple gadgets.

– Even values of k. In this case, N^j is also even and thus, a cycle on N^j+1 vertices may not be part of a bipartite graph. In order to fix that problem, we define $H(e_j)$ as a cycle $\left\{v_1^j, \dots, v_{N^j}^j, v_1^j\right\}$ on N^j vertices, plus an additional edge $[v_{N^j}^j, v_{N^j+1}^j]$. The special vertices l_p^j still are defined as $l_p^j = v_{2k(p-1)+1}^j$ for $p = 1, \cdots, d^j$ (note that $l_{d^j}^j$ never matches $v_{N^j}^j$). Figures 1.2 and 1.3 illustrate $H(e_j)$ for the couple of values $k = 3,\ d^j = 2$ and $k = 4,\ d^j = 2$, respectively.

3) *Linking element gadgets to k-tuple gadgets.* For any couple (e_j, c_i) such that e_j is the value of c_i on the q-th coordinate, the two gadgets $H(c_i)$ and $H(e_j)$ are connected using one of the edges $[a_2^{i,q}, l_{p_i}^j],\ p_i \in \{1, \dots, d^j\}$. The vertices $l_{p_i}^j$ that will be linked to a given gadget $H(c_i)$ must be chosen so that each vertex l_p^j from any gadget $H(e_j)$ will be connected to exactly one gadget $H(c_i)$.

The described construction obviously leads to a graph $G = (V, E)$ that is bipartite, of maximum degree 3, and such that every of the $\mathbf{P}_k$ it contains is chordless. Its number of vertices is $|V| = 3k^2m + (1-k)kn$: consider, on the one hand, that each gadget $H(c_i)$ is a graph on k^2 vertices and, on the other hand, that $\sum_{j=1}^{kn} d^j = km$ (wlog., we may assume that each element e_j appears at least once in $\mathcal{C}$).

1.2.1.3. *Analyzing the obtained instance of* $\mathbf{P}_k$PACKING

Let us define on G some remarkable $\mathbf{P}_k$ packings on the vertex subsets $V(H(c_i))$ and $V(H(e_j))$.

$\mathbf{P}_k$ packings on $V(H(c_i))$, for $i = 1, \dots, m$:

$$\begin{cases} \mathcal{P}^i = \cup_{q=1}^k P^{i,q} \cup \left\{a_1^{i,1}, a_1^{i,2}, \dots, a_1^{i,k}\right\} & \text{with } P^{i,q} = \left\{a_k^{i,q}, \dots, a_2^{i,q}, l_{i,q}\right\} \ \forall q \\ \mathcal{Q}^i = \cup_{q=1}^k Q^{i,q} & \text{with } Q^{i,q} = \left\{a_k^{i,q}, \dots, a_2^{i,q}, a_1^{i,q}\right\} \ \forall q \end{cases}$$

(where $l_{i,q}$ denotes the vertex from some $H(e_j)$ that is linked to $a_2^{i,q}$)

$\mathbf{P}_k$ packings on $V(H(e_j))$, for $j = 1, \dots, kn$:

$\forall p = 1, \dots, d^j$, $\mathcal{P}_p^j$ is defined as the only possible $\mathbf{P}_k$ partition of $V(H(e_j))\backslash\{l_p^j\}$

Note that these collections are of size $|\mathcal{P}^i| = k+1 \ \forall i$, $|\mathcal{Q}^i| = k \ \forall i$ and $|\mathcal{P}_p^j| = 2d^j - 1 \ \forall j \ \forall p \in \{1, \dots, d^j\}$. With the help of these packings, we now put to the fore three properties that will be the key of our further argumentation.

PROPERTY 1.–

(i) *For any* i*,* $\mathcal{P}^i$ *and* $\mathcal{Q}^i$ *are the only two possible* $\mathbf{P}_k$ *partitions of* $V(H(c_i))$*.*

(ii) *Within a* $\mathbf{P}_k$ *partition of* V*, and for any* $j = 1, \dots, kn$*, the collections* $\mathcal{P}_1^j, \dots, \mathcal{P}_{d^j}^j$ *are the only possible* $\mathbf{P}_k$ *partitions of* $V(H(e_j))$*.*

(iii) *Let* $\mathcal{P}^*$ *be a maximum* $\mathbf{P}_k$ *packing on* G *; we can always assume the following :*
$(iii.a)$ *for any* i*,* $\mathcal{P}^*$ *contains either the packing* $\mathcal{P}^i$*, or the packing* $\mathcal{Q}^i$ *;*
$(iii.b)$ *for any* j*,* $\mathcal{P}^*$ *contains one of the packings* $\mathcal{P}_p^j$*, for some* p*.*

PROOF.– For sake of simplicity, we assume that k is odd, even though the arguments also hold for even values of k.

For (i). Quite immediate, from the observation that a given vertex $a_k^{i,q}$ may only be covered by either $P^{i,q}$ or $Q^{i,q}$.

For (ii). Let $\mathcal{P}$ be a $\mathbf{P}_k$ partition of V and consider an element e_j ; since $H(e_j)$ contains $N^j = k(2d^j - 1) + 1$ vertices, at least one edge e of some P_ℓ in $\mathcal{P}$ links $H(e_j)$ to a given $H(c_i)$, using an l_p^j vertex ; we deduce from the previous point that P_ℓ is some $P^{i,q}$ path and thus, that l_p^j is the only vertex of P_ℓ that intersects $H(e_j)$. Consider now any two vertices l_p^j and $l_{p'}^j$, $p < p'$, from $H(e_j)$; the $2k(p' - p) - 1$

vertices that separate l_p^j and $l_{p'}^j$ might not be covered by any collection of $\mathbf{P}_k$. Hence, exactly one l_p^j vertex of $H(e_j)$ is covered by some $P^{i,q}$ and thus, $\mathcal{P}$ contains the corresponding $\mathbf{P}_k$ packing $\mathcal{P}_p^j$.

For $(iii.a)$. Any maximal size $\mathbf{P}_k$ packing must use (at least) one of the two vertices $a_1^{i,q}$ and $l_{i,q}$, for any couple (i,q), where $l_{i,q}$ denotes the vertex from some $H(e_j)$ that is linked to $a_2^{i,q}$. Suppose the reverse, for some (i,q) : then, none of the vertices $l_{i,q}, a_1^{i,q}, a_2^{i,q}, \ldots, a_k^{i,q}$ may be part of a path from $\mathcal{P}^*$ and thus, $P^{i,q}$ or $Q^{i,q}$ could be added to $\mathcal{P}^*$, that would contradict the optimality of $\mathcal{P}^*$. If the edge $[a_1^{i,q}, a_2^{i,q}]$ (*resp.*, $[a_2^{i,q}, l_{i,q}]$ and not $[a_1^{i,q}, a_2^{i,q}]$) is used by some path $P \in \mathcal{P}^*$, then P can be replaced in $\mathcal{P}^*$ by the path $Q^{i,q}$ (*resp.*, by $P^{i,q}$). If none of the edges $[a_1^{i,q}, a_2^{i,q}]$ and $[a_2^{i,q}, l_{i,q}]$ are used by $\mathcal{P}^*$, replace by $P^{i,q}$ (*resp.*, by $Q^{i,q}$) the path from $\mathcal{P}^*$ that uses $l_{i,q}$ (*resp.*, $a_1^{i,q}$ and not $l_{i,q}$). At that point, the collection $\mathcal{P}^*$ contains for any k-tuple c_i at least k paths $P^{i,q}$ and $Q^{i,q}$ (one for each coordinate $q = 1, \ldots, k$). Now, each time $\mathcal{P}^*$ does not contain the packing $\mathcal{P}^i$, we replace these paths by the whole collection $\mathcal{Q}^i$. ■

For $(iii.b)$. Assume the reverse, for some element e_j ; that means that at least 2 vertices $l_{p_i}^j$ and $l_{p_{i'}}^j$ of $H(e_j)$ are used in $\mathcal{P}^*$ by paths $P^{i,q}$ and $P^{i',q'}$, with $p_i < p_{i'}$ (or $\mathcal{P}^*$ would not be of maximal size). Choose two consecutive such vertices, in the sense that $\mathcal{P}^*$ does not use any of the paths $P^{i'',q''}$ for $l_{p_{i''}}^j$ such that $p_i < p_{i''} < p_{i'}$. Since there are $2k(p_{i'} - p_i) - 1$ vertices of $H(e_j)$ between $l_{p_i}^j$ and $l_{p_{i'}}^j$, we can replace $P^{i,q}$, $P^{i',q'}$ and the paths of $\mathcal{P}^*$ between vertices $l_{p_i}^j$ and $l_{p_{i'}}^j$ by $P^{i,q}$ and $2(p_{i'} - p_i)$ paths using vertices between $l_{p_i}^j$ and $l_{p_{i'}}^j$, plus $l_{p_{i'}}^j$. Observe that, in such a case, the packing $\mathcal{P}^{i'}$ will be replaced in $\mathcal{P}^*$ by the packing $\mathcal{Q}^{i'}$, according to the previous property. By repeating this procedure, we obtain a maximal size $\mathbf{P}_k$ packing that fulfills the requirements of items $(iii.a)$ and $(iii.b)$.

1.2.1.4. ***NP**-completeness and **APX**-hardness*

THEOREM 1.– $\mathbf{P}_k$PARTITION *and* INDUCED $\mathbf{P}_k$PARTITION *are **NP**-complete in bipartite graphs of maximum degree 3, for any $k \geq 3$.*
As a consequence, MAX(INDUCED)$\mathbf{P}_k$PACKING *and* MINk-PATHPARTITION *are **NP**-hard in bipartite graphs with maximum degree 3, for any $k \geq 3$.*

PROOF.– Let $I = (X_1, \ldots, X_k; \mathcal{C})$ and $G = (V, E)$ be an instance of kDM and the graph produced by construction described is Subsection 1.2.1.2, respectively. First, we recall that any path of length $k-1$ in G is chordless; thus, the result holds for both $\mathbf{P}_k$PARTITION and INDUCED $\mathbf{P}_k$PARTITION. We claim that there exists a perfect matching $M \subseteq \mathcal{C}$ on I *iff* there exists a partition $\mathcal{P}$ of G into $\mathbf{P}_k$.

Let $\mathcal{P}$ be such a partition on G; from Property 1 item (i), we know that each gadget $H(c_i)$ is covered by either $\mathcal{P}^i$ or $\mathcal{Q}^i$. Moreover, Property 1 item (ii) indicates that every gadget $H(e_j)$ is covered by some $\mathcal{P}_p^j$ collection; those two facts ensure

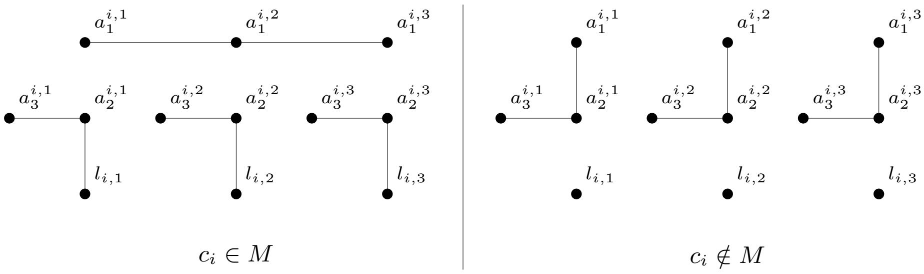


Figure 1.4. *A vertex partition of a $H(c_i)$ gadget into 2-edge paths.*

that exactly one $H(c_i)$ gadget for some k-tuple that contains e_j is covered by a $\mathcal{P}^i$ collection and therefore, the set $M = \left\{c_i \mid \mathcal{P}^i \subseteq \mathcal{P}\right\}$ defines a perfect matching on I.

Conversely, let M be a perfect matching on $\mathcal{C}$; we build a packing $\mathcal{P}$ applying the following rule : if a given element c_i belongs to M, then use $\mathcal{P}^i$ to cover $H(c_i)$; use $\mathcal{Q}^i$ otherwise (Figure 1.4 illustrates this construction for 3DM). Since M is a perfect matching, exactly one vertex l_p^j per gadget $H(e_j)$ is covered by some $P^{i,q}$. Thus, on a given cycle $H(e_j)$, the $N^j = k(2d^j - 1)$ vertices that remain uncovered can be covered using the corresponding collection $\mathcal{P}_p^j$. ■

Thus, the construction is a Karp reduction, and from the **NP**-completeness of kDM, [GAR 79], we deduce the **NP**-completeness of (INDUCED) $\mathbf{P}_k$PARTITION in bipartite graphs of maximum degree 3. However, by a more accurate observation, we actually may obtain a stronger result, for $k = 3$; namely, (INDUCED) $\mathbf{P}_3$PARTITION **NP**-completeness still holds when restricting ourselves to planar instances. Indeed, on the one hand, the restriction PLANAR 3DM of 3-dimensional matching to planar instances still is **NP**-complete, [DYE 86] ; on the other hand, if the initial instance I of kDM is planar, then the graph G also is planar for an appropriate choice of the linking edges $[a_2^{i,q}, l_{i,q}]$.

THEOREM 2.– $\mathbf{P}_3$PARTITION *and* INDUCED $\mathbf{P}_3$PARTITION *are **NP**-complete in planar bipartite graphs with maximum degree 3.*
As a consequence, MAX(INDUCED)$\mathbf{P}_3$PACKING *and* MIN3-PATHPARTITION *are **NP**-hard in planar bipartite graphs with maximum degree 3.*

If we now turn to the optimization problems, we can observe that the construction described in Subsection 1.2.1.2 also enables to establish an **APX**-hardness result for the maximization problems MAX$\mathbf{P}_k$PACKING and MAX (INDUCED) $\mathbf{P}_k$PACKING. We consider the optimization version of kDM, denoted by MAXkDM, and the following inapproximability result : for any $k \geq 3$, there is a constant $\varepsilon'_k > 0$ such that $\forall I = (X_1, \ldots, X_k; \mathcal{C})$ instance of kDM with $|X_1| = \cdots = |X_k| = n$, it is **NP**-hard to decide between $\mathrm{opt}(I) = n$ and $\mathrm{opt}(I) \leq (1 - \varepsilon'_k)n$, where $\mathrm{opt}(I)$ is the value of a maximum matching on $\mathcal{C}$. This result also holds if we restrict ourselves to instances with bounded degree, namely, to instances I satisfying : $\forall j = 1, \ldots, kn,\ d^j \leq f(k)$, where $f(k)$ is a constant ; we refer to [PET 94] for $k = 3$ (where the result is proved with $f(3) = 3$), to [KAR 06] for other values of k.

THEOREM 3.– *For any $k \geq 3$, there is a constant $\varepsilon_k > 0$, such that $\forall G = (V, E)$ instance of* MAX(INDUCED)$\mathbf{P}_k$PACKING *where G is a bipartite graph of maximum degree 3, it is **NP**-hard to decide between* $\mathrm{opt}(G) = \frac{|V|}{k}$ *and* $\mathrm{opt}(G) \leq (1 - \varepsilon_k)\frac{|V|}{k}$, *where* $\mathrm{opt}(G)$ *is the value of a maximum (induced)* $\mathbf{P}_k$*-Packing on G.*

PROOF.– Let $I = (X_1, \ldots, X_k; \mathcal{C})$ be an instance of kDM, with $|X_q| = n\ \forall q$ and $|\mathcal{C}| = m$, such that the degree d^j of any element e_j is bounded above by $f(k)$. Consider the graph $G = (V, E)$ produced by the construction described in Subsection 1.2.1.2 ; we recall that $|V| = 3k^2m - k^2n + kn$. Let (M^*, $\mathcal{P}^*$) be a couple of optimal solutions on I and G, with values $\mathrm{opt}(I)$ and $\mathrm{opt}(G)$, respectively. From Property 1 items $(iii.a)$ and $(iii.b)$, we can assume that $\mathcal{P}^*$ satisfies the following :

– for any i, $\mathcal{P}^*$ contains either the packing $\mathcal{P}^i$, or the packing $\mathcal{Q}^i$;

– for any j, $\mathcal{P}^*$ contains one of the packings $\mathcal{P}^j_1, \ldots, \mathcal{P}^j_{d^j}$.

Hence, the set $M = \{c_i \in \mathcal{C} :\ \mathcal{P}^i \in \mathcal{P}^*\}$ of k-tuples c_i such that $\mathcal{P}^*$ contains $\mathcal{P}^i$ defines a matching on I ; moreover, the value $\mathrm{opt}(G)$ of $\mathcal{P}^*$ can be expressed as :

$$\mathrm{opt}(G) = (km + |M|) + \sum_{j=1}^{kn} \left(2d^j - 1\right) = 3km - kn + |M|$$

From $|M| \leq |M^*|$, we then deduce : $\mathrm{opt}(G) \leq \mathrm{opt}(I) + 3km - kn$.

If $\mathrm{opt}(I) = n$: we know from Theorem 1 that I has a perfect matching *iff* G admits a $\mathbf{P}_k$Partition, that is, $\mathrm{opt}(I) = n$ *iif* $\mathrm{opt}(G) = \frac{|V|}{k} = 3km - kn + n$. Suppose now that $\mathrm{opt}(I) \leq (1 - \varepsilon'_k)n$. Then, necessarily : $\mathrm{opt}(G) \leq 3km - kn + (1 - \varepsilon'_k)n = (3km - kn + n) - \varepsilon'_k n$. By setting $\varepsilon_k = \frac{n}{3km - kn + n}\varepsilon'_k$, we obtain $\mathrm{opt}(G) \leq (1 - \varepsilon_k)(3km - kn + n)$. Finally, since $d^j \leq f(k)$, we deduce that $km \leq kf(k)n$ and then, that $\varepsilon_k \geq \frac{1}{3f(k)k - k + 1}\varepsilon'_k = \mathcal{O}(1)$. In conclusion, deciding between $\mathrm{opt}(G) = |V|/n$ and $\mathrm{opt}(G) \leq (1 - \varepsilon_k)|V|/n$ (or $\mathrm{opt}(G) \leq (1 - \frac{1}{3f(k)k - k + 1}\varepsilon'_k)|V|/n)$) on G would enable to decide between $\mathrm{opt}(I) = n$ and $\mathrm{opt}(I) \leq (1 - \varepsilon'_k)n$ on I. ■

1.2.2. *Positive results from the maximum independent set problem*

If we decrease the maximum degree of the graph down to 2, we can easily prove that $\mathbf{P}_k$PARTITION, INDUCED $\mathbf{P}_k$PARTITION, MAX$\mathbf{P}_k$PACKING and MINk-PATH-PARTITION are polynomial-time computable. The same fact holds for MAXW$\mathbf{P}_k$PACKING (what remains true in forests), although it is a little bit complicated : the proof consists of a reduction from MAXW$\mathbf{P}_k$PACKING in graphs with maximum degree 2 (*resp.*, in a forest) to the problem of computing a maximum weight independent set in an interval (*resp.*, a chordal) graph, which is known to be polynomial, [FRA 76].

PROPOSITION 1.– MAXW$\mathbf{P}_k$PACKING *is polynomial in graphs with maximum degree 2 and in forests, for any* $k \geq 3$.

PROOF.– Let $I = (G, w)$ be an instance of MAXW$\mathbf{P}_k$PACKING where $G = (V, E)$ is a graph with maximum degree 2. Hence, G is a collection of disjoint paths or cycles and thus, each connected component may be separately solved. Moreover, wlog., we may assume that each connected component G^ℓ of G is a path. Otherwise, a given cycle $G^\ell = \{v_1, \ldots, v_{N_\ell}, v_1\}$ might be solved by picking the best solution among the solutions computed on the k instances $G^\ell \setminus \{[v_1, v_2]\}, \ldots, G^\ell \setminus \{[v_k, v_{k+1}]\}$. Thus, let $G^\ell = \left\{v_1^\ell, \ldots, v_{N^\ell}^\ell\right\}$ be such a path ; we build the instance (H^ℓ, w^ℓ) of MAXWIS where the vertex set of H^ℓ corresponds to the paths of length $k-1$ in G^ℓ : a vertex v is associated to each path P_v, with weight $w^\ell(v) = w(P_v)$. Moreover, two vertices $u \neq v$ are linked in H^ℓ *iff* the corresponding paths P_u and P_v share at least one common vertex in the initial graph. We deduce that the set of independent sets in H^ℓ corresponds to the set of $\mathbf{P}_k$ in G^ℓ. Observe that H^ℓ is an interval graph (even a unit interval graph), since each path can be viewed as an interval of the line $\{1, \cdots, N^\ell\}$; hence, H^ℓ is chordal. If G is a forest, then any of the graphs H^ℓ that correspond to a tree of G is a chordal graph. ■

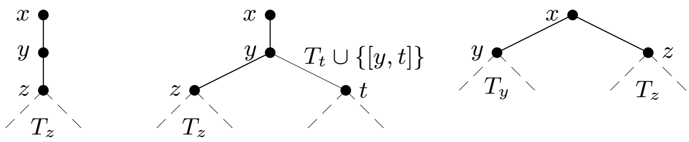
1.3. Approximating MAXWP$_3$PACKING and MIN3-PATHPARTITION

We present some approximation results for MAXW$\mathbf{P}_3$PACKING and MIN3-PATHPARTITION, that are mainly based on matching and spanning tree heuristics.

1.3.1. MAXWP$_3$PACKING *in graphs of maximum degree 3*

For this problem, the best approximate algorithm known so far provides a ratio of $(\frac{1}{2} - \varepsilon)$, within high (but polynomial) time complexity. This algorithm is deduced from the one proposed in [ARK 98] to approximate the weighted k-set packing problem for sets of size 3. Furthermore, a simple greedy $1/k$-approximation of MAXW$\mathbf{P}_k$PACKING consists in iteratively picking a path of length $k-1$ that is of maximum weight. For $k = 3$ and in graphs of maximum degree 3, the time complexity of this algorithm is between $\mathcal{O}(n \log n)$ and $\mathcal{O}(n^2)$ (depending on the encoding structure). Actually, in such graphs, one may reach a $1/3$-approximate solution, even in time $\mathcal{O}(\alpha(n, m)n)$, where α is the inverse Ackerman's function and $m \leq 3n/2$.

THEOREM 4.– MAXW$\mathbf{P}_3$PACKING *is 1/3 approximable within* $\mathcal{O}(\alpha(n, 3n/2)n)$ *time complexity in graphs of maximum degree 3 ; this ratio is tight for the algorithm we analyze.*



Steps 3.1 and 3.2 Steps 3.1, 3.2 and 3.3.1 Steps 4.1 to 4.3

Figure 1.5. *The main configurations of the algorithm* `SubProcess`.

PROOF.– The argument uses the following observation : for any spanning tree of maximum degree 3 containing at least 3 vertices, one can build a cover of its edge set into 3 packings of $\mathbf{P}_3$ within linear time. Hence, by computing a maximum-weight spanning tree $T = (V, E_T)$ on G in $\mathcal{O}(\alpha(n, 3n/2)n)$ time, [CHA 00], and by picking the best $\mathbf{P}_3$-packing among the cover, we obtain a 1/3 approximate solution within an overall time complexity dominated by $\mathcal{O}(\alpha(n, 3n/2)n)$.

The construction of the 3 packings $\mathcal{P}^1, \mathcal{P}^2, \mathcal{P}^3$ is done in the following way : we start with three empty collections $\mathcal{P}^1, \mathcal{P}^2, \mathcal{P}^3$ and a tree T rooted at r ; according to the degree of r and to the degree of its children, we add some $\mathbf{P}_3$ path P that contains r to the packing $\mathcal{P}^1$, remove the edges of P from T, and then recursively repeat this process on the remaining subtrees, alternatively invoking $\mathcal{P}^2$ and $\mathcal{P}^1$. This procedure is formally described in the algorithms `SubProcess` (the recursive process) and `Tree-P`$_3$`PackingCover` (the whole process).

Algorithm `Tree-P`$_3$`PackingCover` makes an initial call to `SubProcess`, on the whole tree T, rooted on a vertex r that is of degree at most 2 in T. The stopping criterion of the recursive procedure `SubProcess` are the following : the current tree has no edge (then stop), or the current tree is a lonely edge $[x, y]$; then add $\{r_x, x, y\}$ to $\mathcal{P}^3$, where r_x denotes the father of x in T. Concerning the three main configurations of `SubProcess`, they are illustrated in Figure 1.5, where T_v denotes the subtree of T rooted at v ; the edges in rigid lines represent the path that is added to the current packing, and the subtrees that are invoked by the recursive calls are indicated.

`Tree-P`$_3$`PackingCover`

Input : $T = (V_T, E_T)$ spanning tree of maximum degree 3 containing at least 3 vertices and rooted at r such that $d_T(r) \leq 2$.

1 Set $\mathcal{P}^1 = \mathcal{P}^2 = \mathcal{P}^3 = \emptyset$;

2 Call `SubProcess`(T_r,$\mathcal{P}^1$,$\mathcal{P}^2$,$\mathcal{P}^3$,1) ;

3 `Repair`($\mathcal{P}^1, \mathcal{P}^2, \mathcal{P}^3$) ;

Output ($\mathcal{P}^1$, $\mathcal{P}^2$, $\mathcal{P}^3$).

```
SubProcess(T_x, P^1, P^2, P^3, i)
    1 If E_{T_x} = ∅ then exit;
       Pick y a child of x in T_x;
    2 If E_{T_x} = {{x, y}}
         Pick r_x the father of x in T_r;
  2.1    P^3 ⟵ P^3 ∪ {{r_x, x, y}}; exit;
    3 If x is of degree 1 in T_x
         Pick z a child of y in T_x;
  3.1    P^i ⟵ P^i ∪ {{x, y, z}};
  3.2    Call SubProcess(T_z, P^1, P^2, P^3, 3-i);
  3.3    If y is of degree 3 in T_x
           Pick t the second child of y in T_x;
 3.3.1     Call SubProcess({{y, t}} ∪ T_t, P^1, P^2, P^3, 3-i);
    4 Else If x is of degree 2 in T_x
         Pick z the second child of x in T_x;
  4.1    P^i ⟵ P^i ∪ {{y, x, z}};
  4.2    Call SubProcess(T_y, P^1, P^2, P^3, 3-i);
  4.3    Call SubProcess(T_z, P^1, P^2, P^3, 3-i);
```

At the end of the initial call to `SubProcess` (that is, when the step 2 of `Tree-P`$_3$`PackingCover` has been achieved), $\mathcal{P}^1$ and $\mathcal{P}^2$ both are packings : one can easily see that the paths that are added to $\mathcal{P}^i$ (where $i = 1$ or $i = 2$) at a given time t and the ones that are added again to $\mathcal{P}^i$ at time $t + 2$ do not share any common vertex. On the other hand, $\mathcal{P}^3$ might not be a packing. Let $\{r_x, x, y\}$ and $\{r_{x'}, x', y'\}$ be two paths from $\mathcal{P}^3$ such that $\{r_x, x, y\} \cap \{r_{x'}, x', y'\} \neq \emptyset$; then, either $r_x = r_{x'}$, or $r_x = x'$. If the first case occurs, $\{x, r_x, x'\}$ has been added to $\mathcal{P}^i$ (for $i = 1$ or $i = 2$), then set : $\mathcal{P}^i = \mathcal{P}^i \backslash \{\{x, r_x, x'\}\} \cup \{\{r_x, x, y\}\}$ and $\mathcal{P}^3 = \mathcal{P}^3 \backslash \{\{r_x, x, y\}\}$. Otherwise, $r_{x'}$ is the father of r_x in T_r and we have $\{r_{x'}, r_x, x\} \in \mathcal{P}^i$ (for $i = 1$ or $i = 2$); then set : $\mathcal{P}^i = \mathcal{P}^i \backslash \{\{r_{x'}, r_x, x\}\} \cup \{\{r_{x'}, x', y'\}\}$ and $\mathcal{P}^3 = \mathcal{P}^3 \backslash \{\{r_{x'}, x', y'\}\}$. These repairing operations are made by the algorithm `Repair`, during step 3 of `Tree-P`$_3$`PackingCover`.

Figure 1.6 provides two examples of the construction of $\mathcal{P}^1$, $\mathcal{P}^2$ and $\mathcal{P}^3$. The overall time complexity of `Tree-P`$_3$`PackingCover` is linear : first, the number of recursive calls to `SubProcess` may not exceed $2/3n$ and second, $|\mathcal{P}^3|$ is at most $\mathcal{O}(\log n)$.

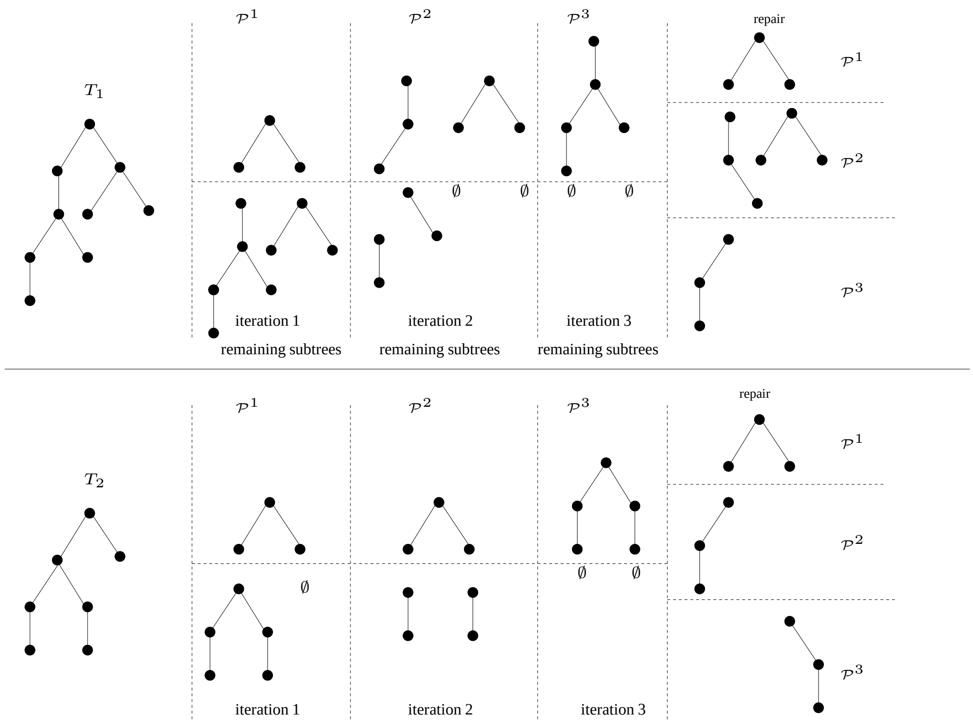


Figure 1.6. *Two examples of the construction of the 3 packings $\mathcal{P}^i$ for $i = 1, 2, 3$.*

`Repair`($\mathcal{P}^1$,$\mathcal{P}^2$,$\mathcal{P}^3$)

1 For any $(P = \{r_x, x, y\} \neq P' = \{r_{x'}, x', y'\}) \in \mathcal{P}^3$ s.t. $r_x = r_{x'}$
Set $i \in \{1, 2\}$ s.t. $\{x, r_x, x'\} \in \mathcal{P}^i$;
1.1 $\mathcal{P}^i \longleftarrow \mathcal{P}^i \backslash \{\{x, r_x, x'\}\} \cup \{\{r_x, x, y\}\};\ \mathcal{P}^3 \longleftarrow \mathcal{P}^3 \backslash \{\{r_x, x, y\}\};$
2 For any $(P = \{r_x, x, y\} \neq P' = \{r_{x'}, x', y'\}) \in \mathcal{P}^3$ s.t. $r_x = x'$
Set $i \in \{1, 2\}$ s.t. $\{r_{x'}, r_x, x\} \in \mathcal{P}^i$;
2.1 $\mathcal{P}^i \longleftarrow \mathcal{P}^i \backslash \{\{r_{x'}, r_x, x\}\} \cup \{\{r_{x'}, x', y'\}\};\ \mathcal{P}^3 \longleftarrow \mathcal{P}^3 \backslash \{\{r_{x'}, x', y'\}\};$
Output ($\mathcal{P}^1$, $\mathcal{P}^2$, $\mathcal{P}^3$).

We now can deduce an approximate algorithm MaxW$\mathbf{P}_3$Packing, that consists in computing a $\mathbf{P}_3$-packing cover $(\mathcal{P}^1, \mathcal{P}^2, \mathcal{P}^3)$ of a maximum spanning tree of G, and then picking the best collection among ($\mathcal{P}^1$, $\mathcal{P}^2$, $\mathcal{P}^3$). This algorithm provides a 1/3-approximation within $\mathcal{O}(\alpha(n, 3n/2)n)$ time complexity (the overall complexity of the algorithm is dominated by the one of computing the initial spanning tree). Concerning the approximation level, consider that the weight $w(T)$ of a maximum spanning tree T is at least the weight of an optimal $\mathbf{P}_3$-packing, since any $\mathbf{P}_3$ packing can be completed into a spanning tree (if the input graph is connected). Then the result is trivial (let $\mathcal{P}^*$ denote an optimal solution) :

$$w(\mathcal{P}) \geq 1/3\left(w(\mathcal{P}^1) + w(\mathcal{P}^2) + w(\mathcal{P}^3)\right) \geq 1/3w(T) \geq 1/3w(\mathcal{P}^*)$$

The proof of tightness is omitted. ■

1.3.2. MAXW$\mathbf{P}_3$PACKING *in bipartite graphs of maximum degree 3*

If we restrict ourselves to bipartite graphs, we slightly improve the ratio of $\frac{1}{2} - \varepsilon$, [ARK 98] up to $\frac{1}{2}$. We then show that, in the unweighted case, this result holds without any constraint on the graph maximum degree. The key idea here is to transform the problem of finding a $\mathbf{P}_3$Packing in the initial bipartite graph $G = (L, R; E)$ into the problem of finding a maximum matching in two graphs G_L and G_R, where G_L (*resp.*, G_R) contains the representative edge of the $\mathbf{P}_3$ of the initial graph with their two extremities in L (*resp.*, in R). Formally, from an instance $I = (G, w)$ of MAXW$\mathbf{P}_3$PACKING, where $G = (L, R; E)$ is a bipartite graph of maximum degree 3, we build two weighted graphs (G_L, w_L) and (G_R, w_R), where $G_L = (L, E_L)$ and $G_R = (R, E_R)$. Two vertices $x \neq y$ from L are linked in G_L *iff* there exists in G a path $P_{x,y}$ of length 2 from x to y : $[x, y] \in E_L$ *iff* $\exists z \in R$ s.t. $[x, z], [z, y] \in E$. The weight $w_L(x, y)$ is defined as $w_L(x, y) = \max\{w(x, z) + w(z, y) | [x, z], [z, y] \in E\}$. The weighted graph (G_R, w_R) is defined by considering R instead of L. If G is of maximum degree 3, then the following fact holds :

PROPERTY 2.– *From any matching M on G_L (*resp.*, on G_R), one can deduce a $\mathbf{P}_3$ packing $\mathcal{P}_M$ of weight $w(\mathcal{P}_M) = w_L(M)$ (*resp.*, $w(\mathcal{P}_M) = w_R(M)$), where G is of degree at most 3.*

PROOF.– Let M be a matching on G_L, and $\mathcal{P}_M$ the corresponding $\mathbf{P}_3$ collection on G. Suppose that two paths $P_{x,y} \neq P_{x',y'} \in \mathcal{P}_M$ share a common vertex t. Because M is a matching, we have $\{x, y\} \cap \{x', y'\} = \emptyset$; hence, the vertex t belongs to R and is the internal vertex of both $P_{x',y'}$ and $P_{x',y'}$, which contradicts the assumption on the graph maximum degree. ■

In light of this fact, we propose the algorithm `Weighted P`$_3$`-Packing` that consists in computing two maximum matchings on G_L and G_R, and then picking the best corresponding packing in G. The time complexity of this algorithm is mainly the time complexity of computing a maximum weight matching in graphs of maximum degree 9, that is $\mathcal{O}(|V|^2 \log |V|)$, [LOV 86].

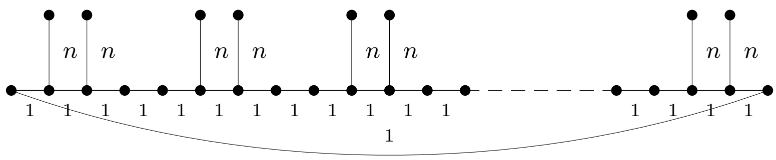


Figure 1.7. *Tightness of `Weighted P`$_3$`-Packing` analysis.*

`Weighted P`$_3$`-Packing`

1 Build the weighted graphs (G_L, w_L) and (G_R, w_R) ;

2 Compute a maximum weight matching M_L^* (*resp.*, M_R^*) on (G_L, w_L) (*resp.*, on (G_R, w_R)) ;

3 Deduce from M_L^* (*resp.*, from M_R^*) a $\mathbf{P}_3$ packing $\mathcal{P}_L$ (*resp.*, $\mathcal{P}_R$) according to Property 2 ;

4 Output the best packing $\mathcal{P}$ among $\mathcal{P}_L$ and $\mathcal{P}_R$.

THEOREM 5.– *`Weighted P`$_3$`-Packing` provides a $1/2$-approximation for* MAXW$\mathbf{P}_3$-PACKING *in bipartite graphs with maximum degree 3 and this ratio is tight.*

PROOF.– Let $\mathcal{P}^*$ be an optimum $\mathbf{P}_3$-packing on $I = (G, w)$, we denote by $\mathcal{P}_L^*$ (*resp.*, by $\mathcal{P}_R^*$) the paths of $\mathcal{P}^*$ of which the two endpoints belong to L (*resp.*, to R) ; thus, $\text{opt}(I) = w(\mathcal{P}_L^*) + w(\mathcal{P}_L^*)$. For any path $P = P_{x,y} \in \mathcal{P}_L^*$, $[x, y]$ is an edge from E_L, of weight $w_L(x, y) \geq w(P_{x,y})$. Hence, $M_L = \{[x, y]|P_{x,y} \in \mathcal{P}_L^*\}$ is a matching on G_L that satisfies :

$$w_L(M_L) \geq w(\mathcal{P}_L^*) \quad [1.1]$$

Moreover, since M_L^* is a maximum weight matching on G_L, we have $w_L(M_L) \leq w_L(M_L^*)$. Thus, using inequality [1.1] and Property 2 (and by applying the same arguments on G_R), we deduce :

$$w(\mathcal{P}_L) \geq w(\mathcal{P}_L^*), \; w(\mathcal{P}_R) \geq w(\mathcal{P}_R^*) \quad [1.2]$$

Finally, the solution output by the algorithm satisfies $w(\mathcal{P}) \geq 1/2(w(\mathcal{P}_L) + w(\mathcal{P}_R))$ and we directly deduce from inequalities [1.2] the expected result. The instance $I = (G, w)$ that provides the tightness is depicted in Figure 1.7. It consists of a graph on $12n$ vertices on which one can easily observe that $w(\mathcal{P}_L) = w(\mathcal{P}_R) = 2n(n+2)$ and $w(\mathcal{P}^*) = 2n(2n+2)$. ■

Concerning the unweighted case, we may obtain the same performance ratio without the restriction on the graph maximum degree. The main differences compared to the previous algorithm lie in the construction of the two graphs G_L, G_R : starting from G, we duplicate each vertex $r_i \in R$ by adding a new vertex r'_i with the same neighborhood as r_i (this operation, often called *multiplication of vertices* in the literature, is used in the characterization of perfect graphs). We then add the edge $[r_i, r'_i]$. If R_L denotes the vertex set $\{r_i, r'_i | r_i \in R\}$, the following properties hold :

PROPERTY 3.–

(i) *From any matching M on G_L, one can deduce a matching M' of cardinality $|M'| \geq |M|$ on G_L that saturates R_L.*

(ii) *From any matching M on G_L (*resp.*, on G_R) that saturates R_L (*resp.*, L_R), one can deduce a $\mathbf{P}_3$ packing $\mathcal{P}_M$ on G of size $|\mathcal{P}_M| = |M| - |R|$.*

PROOF.– *For* (i). Let M be a matching on G_L and consider a given vertex $r_i \in R$. If M contains no edge incident to $\{r_i, r'_i\}$, then add $[r_i, r'_i]$ to M ; if M contains an edge e incident to r_i (*resp.*, to r'_i), but no edge incident to r'_i (*resp.*, to r_i), then set $M = M \backslash \{e\} \cup \{[r_i, r'_i]\}$.

For (ii). Let M be a matching on G_L that saturates R_L, we respectively denote by J the set of vertices $r_i \in R$ such that $[r_i, r'_i] \in M$ and by $p = |J|$ its cardinality. We consider the matching M' deduced from M by deleting the edges $[r_i, r'_i]$; hence, $|M'| = |M| - p$. From the fact that M saturates R_L, we first deduce that $|M| = |R_L| - p = 2|R| - p$; we then observe that, for any vertex $r_i \notin J$, there exists two edges $[l^1_i, r_i]$ and $[l^2_i, r'i]$ in M', that define the $\mathbf{P_3}$ $P_i = \{l^1_i, r_i, l^2_i\}$ of the initial graph G. The collection $\mathcal{P}_M = \cup_{r_i \notin J} \{P_i\}$ obviously is a $\mathbf{P_3}$ packing of size $|M'|/2$ on G. One just has to obverse that $|M'| = 2|R| - 2p = 2(|M| - |R|)$ in order to conclude. ■

`P`$_3$`-Packing`

1 Build the graph G_L (*resp.*, G_R) obtained from G by *multiplication of vertices* on R (*resp.*, on L) ;

2 Compute a maximum size matching M_L (*resp.*, M_R) on G_L(*resp.*, on G_R) ; According to Property 3 item (i), deduce from M_L (*resp.*, from M_R) a maximum size matching M^*_L (*resp.*, M^*_R) that saturates R_L (*resp.*, L_R) ;

3 According to Property 3 item (ii), deduce from M^*_L (*resp.*, from M^*_R) a $\mathbf{P}_3$ packing $\mathcal{P}_L$ (*resp.*, $\mathcal{P}_R$) of size $|M^*_L| - |R|$ (*resp.*, $|M^*_R| - |L|$) ;

4 Output the best packing $\mathcal{P}$ among $\mathcal{P}_L$ and $\mathcal{P}_R$.

The approximate algorithm `P`$_3$`-Packing` works as previously, except that we compute a maximum (size) matching M^*_L (resp., M^*_R) on G_L (*resp.*, G_R) that saturates R_L

(*resp.*, L_R) in step 2, and that the $\mathbf{P}_3$ packing $\mathcal{P}_L$ (*resp.*, $\mathcal{P}_R$) is obtained from M_L^* (*resp.*, M_R^*) by deleting the edges $[r_i, r_i']$ (*resp.*, $[l_i, l_i']$) in step 3.

THEOREM 6.– `P3-Packing` *provides a* $1/2$*-approximation for* MAX$\mathbf{P}_3$PACKING *in bipartite graphs and this ratio is tight. The time complexity of this algorithm is* $\mathcal{O}(m\sqrt{n})$.

PROOF.– Let $\mathcal{P}_L^* = \{P_1, \cdots, P_q\}$ be the set of paths from the optimal solution having their two endpoints in L ; $\mathcal{P}_L^*$ can easily be converted on G_L into a matching M of size $|M| = 2q + (|R| - q) = |\mathcal{P}_L^*| + |R|$. From the optimality of M_L^* on G_L, we deduce that $|M_L^*| \geq |M|$ and hence, that $|\mathcal{P}_L| \geq |\mathcal{P}_L^*|$. The same obviously holds for $\mathcal{P}_R^*$ and the result is immediate. The time complexity of the unweighted version of the algorithm still is dominated by the one of computing a maximum (size) matching, that is $\mathcal{O}(m\sqrt{n})$, [LOV 86]. The proof of tightness is omitted. ■

1.3.3. MIN3-PATHPARTITION *in general graphs*

To our knowledge, the approximability of MINk-PATHPARTITION (or MINPATHPARTITION) has not been studied so far. Here, we propose a $3/2$-approximation for MIN3-PATHPARTITION. Although this problem can be viewed as an instance of 3-set cover (view the set of all paths of length 0, 1 or 2 in G as sets on V), MIN3-PATHPARTITION and the minimum 3-set cover problem are different. For instance, consider a star $K_{1,2n}$; the optimum value of the corresponding 3-set cover instance is n, whereas the optimum value of the 3-path partition is $2n - 1$. Note that, concerning MINPATHPARTITION (that is, the approximation of $\rho(G)$), we can trivially see that it is not $(2 - \varepsilon)$-approximable, from the fact that deciding whether $\rho(G) = 1$ or $\rho(G) \geq 2$ is **NP**-complete. Actually, we can more generally establish that $\rho(G)$ is not in **APX** : otherwise, we could obtain a **PTAS** for the traveling salesman problem with weight 1 and 2 when $\mathrm{opt}(I) = n$, which is not possible, unless **P**=**NP**. The algorithm `Minimum 3Path Partition` we propose runs in two phases : first, it computes a maximum matching M_1^* on the input graph $G = (V, E)$; then, it matches through M_2^* a maximum number of edges from M_1^* to vertices from $V \backslash M_1^*$. Those two matchings define the $\mathbf{P}_3$ and the $\mathbf{P}_2$ of the approximate solution.

THEOREM 7.– `Minimum 3Path Partition` *provides a* $3/2$*-approximation for* MIN3-PATHPARTITION *in general graphs within* $\mathcal{O}(nm + n^2 \log n)$ *time and this ratio is tight.*

PROOF.– Let $G = (V, E)$ be an instance of MIN3-PATHPARTITION. Let $\mathcal{P}^* = (\mathcal{P}_2^*, \mathcal{P}_1^*, \mathcal{P}_0^*)$ and $\mathcal{P}' = (\mathcal{P}_2', \mathcal{P}_1', \mathcal{P}_0')$ respectively be an optimal solution and the approximate 3-path partition on G, where $\mathcal{P}_i^*$ and $\mathcal{P}_i'$ denote for $i = 0, 1, 2$ the set of paths of length i. By construction of the approximate solution, we have :

$$\mathrm{apx}(I) = |V| - |M_1^*| - |M_2^*| \qquad [1.3]$$

Let $V_0 = (V \setminus V(M_1^*)) \setminus \mathcal{P}_0^*$, we consider a subgraph $G_2' = (L, R'; E_2')$ of G_2, where R' and E_2' are defined as : $R' = \{r_v \in R | v \in V_0\}$ and E_2' contains the edge $[l_e, r_v] \in E_2'$ *iff* there is an edge of $\mathcal{P}^*$ that links v to an endpoint of e. By definition of V_0, we deduce that $d_{G_2'}(r_v) \geq 1$ for any $v \in V_0$ (V_0 is an independent set of G). Moreover, we have $d_{G_2'}(l_e) \leq 2$ for any $e \in M_1^*$ (M_1^* is an optimal matching). Thus, we get :

$$|M_2^*| \geq 1/2|R'| = 1/2\left(|V| - 2|M_1^*| - |\mathcal{P}_0^*|\right) \qquad [1.4]$$

From relations [1.3] and [1.4], we deduce :

$$\text{apx}(I) = |V| - |M_1^*| - |M_2^*| \leq 1/2\left(|V| + |\mathcal{P}_0^*|\right) \qquad [1.5]$$

Now, consider the optimal solution. From $|V| = 3|\mathcal{P}_2^*| + 2|\mathcal{P}_1^*| + |\mathcal{P}_0^*|$, we trivially have :

$$\text{opt}(I) = |\mathcal{P}_2^*| + |\mathcal{P}_1^*| + |\mathcal{P}_0^*| \geq 1/3\left(|V| + |\mathcal{P}_0^*|\right) \qquad [1.6]$$

Thus, we obtain the expected result. The proof of tightness is omitted. Concerning the time complexity, we refer again to [LOV 86]. ■

`Minimum 3Path Partition`

1 Compute a maximum matching M_1^* on G ;

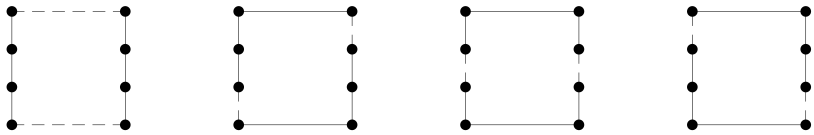
2 Build a bipartite graph $G_2 = (L, R; E_2)$ where $L = \{l_e | e \in M_1^*\}$, $R = \{r_v | v \in V \setminus V(M_1^*)\}$, and $[l_e, r_v] \in E_2$ *iff* the corresponding isolated vertex $v \notin V(M_1^*)$ is adjacent in G to the edge $e \in M_1^*$;

3 Compute a maximum matching M_2^* on G_2 ;

4 Output $\mathcal{P}'$ the 3-paths partition deduced from M_1^*, M_2^*, and $V \setminus V(M_1^* \cup M_2^*)$. Precisely, if $M_1' \subseteq M_1^*$ is the set of edges adjacent to M_2^*, then the paths of length 2 are given by $M_1' \cup M_2^*$, the paths of length 1 are given by $M_1^* \setminus M_1'$, and the paths of length 0 (that is, the isolated vertices) are given by $V \setminus V(M_1^* \cup M_2^*)$.

1.4. Standard and differential approximation of $\mathbf{P}_k$P

From now, we will exclusively deal with the approximability of MAX$\mathbf{P}_k$P and MIN$\mathbf{P}_k$P, from both standard and differential points of view. We recall that $\mathbf{P}_k$P is the special case of MAXW$\mathbf{P}_k$PACKING where the graph is complete on kn vertices. We first discuss the differential approximability of $\mathbf{P}_k$P, for any constant value k, by connection to the differential approximability of the traveling salesman problem. The



$\mathcal{P}_1$ $\mathcal{P}_2$ $\mathcal{P}_3$ $\mathcal{P}_4$

Figure 1.8. *An example of the 4 solutions* $\mathcal{P}_1, \ldots, \mathcal{P}_4$

second part of this Section then focus on the special case where $k = 4$, in the aim of extensively analysing the approximate algorithm proposed by Hassin and Rubinstein, which is described in Paragraph 1.4.2.1. We first considerer, on the one hand, general and metric instances for the standard ratio (Paragraph 1.4.2.2) and, on the other hand, general instances for the differential ratio (Paragraph 1.4.2.3). We then switch to bi-valuated instances, namely : $\{1, 2\}$-instances for the standard ratio (Paragraph 1.4.2.4) and $\{a, b\}$-instances for the differential ratio (Paragraph 1.4.2.5).

1.4.1. *Differential approximation of* $\mathbf{P}_k\mathbf{P}$ *from the traveling salesman problem*

A common technique in order to obtain an approximate solution for MAX$\mathbf{P}_k$P from a Hamiltonian cycle is called the *deleting and turning around* method, see for instance [HAS 97, HAS 06, FRE 78]. Starting from a tour, this method builds k solutions of MAX$\mathbf{P}_k$P and picks the best among them, where the ith solution is obtained by deleting every kth edge from the input cycle, starting from its ith edge. The quality of the output $\mathcal{P}'$ obviously depends on the quality of the initial tour ; in this way, it is proven in [HAS 97, HAS 06], that any ε-standard approximation for MAXTSP provides a $\frac{k-1}{k}\varepsilon$-standard approximation for MAX$\mathbf{P}_k$P. From a differential point of view, things are less optimistic : even for $k = 4$, there exists an instance family $(I_n)_{n\geq 1}$ that verifies $\mathrm{apx}(I_n) = \frac{1}{2}\mathrm{opt}_{\text{MAX}\mathbf{P}_4\text{P}}(I_n) + \frac{1}{2}\mathrm{wor}_{\text{MAX}\mathbf{P}_4\text{P}}(I_n)$. This instance family is defined as $I_n = (K_{8n}, w)$ for $n \geq 1$, where the vertex set $V(K_{8n})$ may be partitioned into two sets $L = \{\ell_1, \ldots, \ell_{4n}\}$ and $R = \{r_1, \ldots, r_{4n}\}$ so that the associated weight function w is 0 on $L \times L$, 2 on $R \times R$ and 1 on $L \times R$. Thus, for any $n \geq 1$, the following property holds :

PROPERTY 4.– $\mathrm{apx}(I_n) = 6n$, $\mathrm{opt}_{\text{MAX}\mathbf{P}_4\text{P}}(I_n) = 8n$, $\mathrm{wor}_{\text{MAX}\mathbf{P}_4\text{P}}(I_n) = 4n$.

PROOF.– If the initial tour is described as $\Gamma = \{e_1, \ldots, e_n, e_1\}$, then the *deleting and turning around* method produces 4 solutions $\mathcal{P}_1, \ldots, \mathcal{P}_4$ where $\mathcal{P}_i = \cup_{j=0}^{n-1}\{\{e_{j+i}, e_{j+i+1}, e_{j+i+2}\}\}$ for $i = 1, \ldots, 4$ (indices are considered mod n). Figure 1.8 provides an illustration of this process (the dashed lines correspond to the edges from $\Gamma \setminus \mathcal{P}_i$).

Observe that any optimal tour Γ on I_n has total weight $8n$ (consider that any tour contains as many edges with their two endpoints in L as edges with their two endpoints in R). Hence, starting from the optimal cycle $\Gamma^* = [r_1, \ldots, r_{4n}, l_1, \ldots, l_{4n}, r_1]$, any

Figure 1.9. *A worst solution and an optimal solution when $n = 1$*

of the four solutions $\mathcal{P}_1, \ldots, \mathcal{P}_4$ output by the algorithm (see Figure 1.8) has value $w(\mathcal{P}_i) = 6n$, while an optimal solution $\mathcal{P}^*$ and a worst solution $\mathcal{P}_*$ are of total weight respectively $8n$ and $4n$ (see Figure 1.9). Indeed, because any $\mathbf{P}_4$-partition $\mathcal{P}$ is a $2n$ edge cut down tour, we get, on the one hand, $\mathrm{opt}_{\text{MAXTSP}}(I_n) \geq w(\mathcal{P})$ and, on the other hand, $w(\mathcal{P}) \geq 8n - 4n = 4n$, which concludes this argument. ∎

Nevertheless, the deleting and turning around method leads to the following weaker differential approximation relation :

LEMMA 1.– *From an ε-differential approximation of* MAXTSP*, one can polynomially compute an $\frac{\varepsilon}{k}$-differential approximation of* MAX$\mathbf{P}_k$P*. In particular, we deduce from [HAS 01, MON 02b] that* MAX$\mathbf{P}_k$P *is $\frac{2}{3k}$-differential approximable.*

PROOF.– Let us show that the following inequality holds for any instance $I = (K_{kn}, w)$ of MAX$\mathbf{P}_k$P :

$$\mathrm{opt}_{\text{MAXTSP}}(I) \geq \frac{1}{k-1}\mathrm{opt}_{\text{MAX}\mathbf{P}_k\text{P}}(I) + \mathrm{wor}_{\text{MAX}\mathbf{P}_k\text{P}}(I) \qquad [1.7]$$

Let $\mathcal{P}^*$ be an optimal solution of MAX$\mathbf{P}_k$P, then arbitrarily add some edges to $\mathcal{P}^*$ in order to obtain a tour Γ. From this latter, we can deduce $k-1$ solutions $\mathcal{P}_i$ for $i = 1, \ldots, k-1$, by applying the deleting and turning around method in such a way that any of the solutions $\mathcal{P}_i$ contains $(\Gamma \setminus \mathcal{P}^*)$. Thus, we get $(k-1)\mathrm{wor}_{\text{MAX}\mathbf{P}_k\text{P}}(I) \leq \sum_{i=1}^{k-1} w(\mathcal{P}_i) = (k-1)w(\Gamma) - \mathrm{opt}_{\text{MAX}\mathbf{P}_k\text{P}}(I)$. Hence, consider that we also have $w(\Gamma) \leq \mathrm{opt}_{\text{MAXTSP}}(I)$ and the result follows. By applying again the deleting and turning around method, but this time from a worst tour, we may obtain k approximate solutions of MAX$\mathbf{P}_k$P, which allows us to deduce :

$$\mathrm{wor}_{\text{MAXTSP}}(I) \geq \frac{k}{k-1}\mathrm{wor}_{\text{MAX}\mathbf{P}_k\text{P}}(I) \qquad [1.8]$$

Finally, let Γ' be an ε-differential approximation of MAXTSP, we deduce from Γ' k approximate solutions of $\text{MAX}\mathbf{P}_k\text{P}$. If $\mathcal{P}'$ is set to the best one, we get $w(\mathcal{P}') \geq \frac{k}{k-1}w(\Gamma')$ and thus :

$$\text{apx}(I) \geq \frac{k}{k-1}w(\Gamma') \geq \frac{k}{k-1}\big(\varepsilon\text{opt}_{\text{MAXTSP}}(I)+(1-\varepsilon)\text{wor}_{\text{MAXTSP}}(I)\big) \quad [1.9]$$

Using inequalities [1.7], [1.8] and [1.9], we get $\text{apx}(I) \geq \frac{\varepsilon}{k}\text{opt}_{\text{MAX}\mathbf{P}_k\text{P}}(I) + (1 - \frac{\varepsilon}{k})\text{wor}_{\text{MAX}\mathbf{P}_k\text{P}}(I)$ and the proof is complete. ■

To conclude with the relationship between $\mathbf{P}_k\text{P}$ and TSP with respect to their approximability, observe that the minimization case with respect to standard approximation also is trickier. Notably, if we consider $\text{MINMETRIC}\mathbf{P}_4\text{P}$, then the instance family $I'_n = (K_{8n}, w')$ built as the same as I_n with a distinct weight function defined as $w'(\ell_i, \ell_j) = w'(r_i, r_j) = 1$ and $w'(\ell_i, r_j) = n^2 + 1$ for any i, j, then we have : $\text{opt}_{\text{TSP}}(I'_n) = 2n^2 + 8n$ whereas $\text{opt}_{\mathbf{P}_4\text{P}}(I'_n) = 6n$.

1.4.2. *Approximating $\mathbf{P}_4\mathbf{P}$ by means of optimal matchings*

Here starts the analysis, from both a standard and a differential point of view, of an algorithm proposed by Hassin and Rubinstein in [HAS 97], where the authors show that the approximate solution is a 3/4-standard approximation for $\text{MAX}\mathbf{P}_4\text{P}$. We prove that, with respect to the standard ratio, this algorithm provides new approximation ratios for $\text{METRIC}\mathbf{P}_4\text{P}$, namely : the approximate solution respectively achieves a 3/2, a 7/6 and a 9/10-standard approximation for $\text{MINMETRIC}\mathbf{P}_4\text{P}$, $\text{MIN}\mathbf{P}_4\text{P}_{1,2}$ and $\text{MAX}\mathbf{P}_4\text{P}_{1,2}$. As a corollary of a more general result, we also obtain an alternative proof of the result of [HAS 97]. We then prove that, under differential ratio, the approximate solution is a 1/2-approximation for general $\mathbf{P}_4\text{P}$ and a 2/3-approximation for $\mathbf{P}_4\text{P}_{a,b}$. In addition to the new approximation bounds that they provide, the obtained results establish the robustness of the algorithm that is addressed here, since this latter provides good quality solutions, whatever version of the problem we deal with, whatever approximation framework within which we estimate the approximate solutions.

Note that the gap between differential and standard approximation levels that might be reached for a maximization problem comes from the fact that, within the differential framework, the approximate value is located within the tighter interval $[\text{wor}(I), \text{opt}(I)]$, instead of $[0, \text{opt}(I)]$ for the standard measure. That is the aim of differential approximation : the reference it does to $\text{wor}(I)$ makes this measure both more precise (relevant with respect to the notion of guaranteed performance) and more robust (in the sense that minimizing and maximizing turn to be equivalent and, more generally, differential ratio is invariant under affine transformation of the objective function).

1.4.2.1. *Description of the algorithm*

The algorithm proposed in [HAS 97] runs in two stages : first, it computes an optimum weight perfect matching M on $I = (K_{4n}, w)$; then, it builds on the edges of M a second optimum weight perfect matching R in order to complete the solution (note that "optimum weight" signifies "*maximum weight*" if the goal is to maximize, "*minimum weight*" if the goal is to minimize). Precisely, we define the instance $I' = (K_{2n}, w')$ (having a vertex v_e in K_{2n} for each edge $e \in M$), where the weight function w' is defined as follows : for any edge $[v_{e_1}, v_{e_1}]$ on I', $w'(v_{e_1}, v_{e_2})$ is set to the weight of the heaviest edge that links e_1 and e_2 in I, that is, if $e_1 = [x_1, y_1]$ and $e_2 = [x_2, y_2]$, then $w'(v_{e_1}, v_{e_2}) = \max\{w(x_1, x_2), w(x_1, y_2), w(y_1, x_2), w(y_1, y_2)\}$ (when dealing with the minimization version of the problem, set the weight to the lightest). We thus build on (K_{2n}, w') an optimum weight matching R, which is then transposed to the initial graph (K_{4n}, w) by selecting on K_{4n} the edges that realizes the same weight. Since the computation of an optimum weight perfect matching is polynomial, the whole algorithm runs in polynomial time, whether the goal is to minimize or to maximize.

1.4.2.2. *General* $\mathbf{P}_4$P *within the standard framework*

For any solution $\mathcal{P}$, we denote respectively by $M_{\mathcal{P}}$ and $R_{\mathcal{P}}$ the set of the end edges and the set of the middle edges of its paths. Furthermore, we consider for any path $P = \{x, y, z, t\}$ of the solution the edge $[t, x]$ that completes P into a cycle. If $\overline{R}_{\mathcal{P}}$ denotes the set of these edges, we observe that $R_{\mathcal{P}} \cup \overline{R}_{\mathcal{P}}$ forms a perfect matching. Finally, for any edge $e \in \mathcal{P}$, we will denote by $P_{\mathcal{P}}(e)$ the $\mathbf{P}_4$ from the solution that contains e and by $C_{\mathcal{P}}(e)$ the 4-edge cycle that contains $P_{\mathcal{P}}(e)$.

LEMMA 2.– *For any instance $I = (K_{4n}, w)$ with optimal solution $\mathcal{P}^*$ and for any perfect matching M, there exist four pairwise disjoint edge sets A, B, C and D that verify :*

(i) $A \cup B = \mathcal{P}^*$ *and* $C \cup D = \overline{R}_{\mathcal{P}^*}$.

(ii) $A \cup C$ *and* $B \cup D$ *both are perfect matchings on* I.

(iii) $A \cup C \cup M$ *is a perfect 2-matching on* I *whose cycles are of length a multiple of 4.*

PROOF.– Let $\mathcal{P}^* = M_{\mathcal{P}^*} \cup R_{\mathcal{P}^*}$ be an optimal solution, we apply the `Combining perfect matchings` process. At the initialization stage, the connected components of the partial graph induced by $(A \cup C \cup M)$ are either cycles that alternate edges from $(A \cup C)$ and M, or isolated edges from $M_{\mathcal{P}^*} \cap M$. During step 2, at each iteration, the process merges together two connected components of G' into a single cycle that still alternates edges from $(A \cup C)$ and M (an illustration of this merging process is provided in Figure 1.10). Note that all along the process, the sets A, B, C and D define a partition of $\mathcal{P}^* \cup \overline{R}_{\mathcal{P}^*}$ and thus, remain pairwise disjoint.

`Combining perfect matchings`

1 $A \longleftarrow M_{\mathcal{P}^*}$, $B \longleftarrow R_{\mathcal{P}^*}$, $C \longleftarrow \emptyset$, $D \longleftarrow \overline{R}_{\mathcal{P}^*}$;

Set $G' = (V, A \cup M)$ (consider the simple graph) ;

2 While $\exists\, e \in R_{\mathcal{P}^*}$ that links two connected components of G', do :

$$2.1 \left| \begin{array}{l} A \longleftarrow A \backslash \left(C_{\mathcal{P}^*}(e) \cap M_{\mathcal{P}^*}\right),\ B \longleftarrow B \cup \left(C_{\mathcal{P}^*}(e) \cap M_{\mathcal{P}^*}\right); \\ B \longleftarrow B \backslash \left(C_{\mathcal{P}^*}(e) \cap R_{\mathcal{P}^*}\right),\ A \longleftarrow A \cup \left(C_{\mathcal{P}^*}(e) \cap R_{\mathcal{P}^*}\right); \\ D \longleftarrow D \backslash \left(C_{\mathcal{P}^*}(e) \cap \overline{R}_{\mathcal{P}^*}\right),\ C \longleftarrow C \cup \left(C_{\mathcal{P}^*}(e) \cap \overline{R}_{\mathcal{P}^*}\right); \end{array} \right.$$

2.2 $G' \longleftarrow (V, A \cup C \cup M)$;

3 Output A, B, C and D.

• For (i) : Immediate from definition of the process (edges from $\mathcal{P}^*$ are moved from A to B, from B to A, but never out of $A \cup B$; the same holds for $\overline{R}_{\mathcal{P}^*}$ and the two sets C and D).

• For (ii) : At the initialization stage, $A \cup C$ and $B \cup D$ respectively coincide with $M_{\mathcal{P}^*}$ and $R_{\mathcal{P}^*} \cup \overline{R}_{\mathcal{P}^*}$, each a perfect matching. More precisely, for any path $P \in \mathcal{P}^*$, if $C(P)$ denotes the associated 4-edge cycle, then $A \cup C$ and $B \cup D$ respectively contain the perfect matching $C(P) \cap M_{\mathcal{P}^*}$ and $C(P) \cap (R_{\mathcal{P}^*} \cup \overline{R}_{\mathcal{P}^*})$ on $V(P)$. Now, at each iteration, the algorithm swaps the perfect matchings that are used in $A \cup C$ or in $B \cup D$ in order to cover the vertices of a given path P and thus, both $A \cup C$ and $B \cup D$ remain perfect matchings.

• For (iii) : At the end of the process, the stopping criterion ensures that $(A \cup C) \cap M = \emptyset$ and thus, as the union of two perfect matchings, $A \cup C \cup M$ is a perfect 2-matching. Now, consider a cycle Γ of $G' = (V, A \cup C \cup M)$; by definition of step 2, any edge e from $R_{\mathcal{P}^*}$ that is incident to Γ has its two endpoints in $V(\Gamma)$, which means that Γ contains either the two edges of $C_{\mathcal{P}^*}(e) \cap M_{\mathcal{P}^*}$, or the two edges of $C_{\mathcal{P}^*}(e) \cap (R_{\mathcal{P}^*} \cup \overline{R}_{\mathcal{P}^*})$. In other words, if any vertex u from any path $P \in \mathcal{P}^*$ belongs to $V(\Gamma)$, then the whole vertex set $V(P)$ actually is a subset of $V(\Gamma)$ and therefore, we deduce that $|V(\Gamma)| = 4q$, where q is the number of paths $P \in \mathcal{P}^*$ such that Γ contains $V(P)$. ■

THEOREM 8.– *The solution $\mathcal{P}'$ provided by the algorithm achieves a 3/2-standard approximation for* MINMETRIC$\mathbf{P}_4$P *and this ratio is tight.*

PROOF.– Let $\mathcal{P}^*$ be an optimal solution on $I = (K_{4n}, w)$. Using Lemma 2 with the perfect matching $M_{\mathcal{P}'}$ of the solution $\mathcal{P}'$, we obtain four pairwise disjoint sets A, B, C and D. According to property (iii), we can split $A \cup C$ into two sets A_1 and A_2 so that $A_i \cup M_{\mathcal{P}'}$ ($i = 1, 2$) is a $\mathbf{P}_4$-partition (see Figure 1.11 for an illustration). Hence, A_i constitutes an alternative solution for $R_{\mathcal{P}'}$ and because this latter is optimal on $I' = (K_{2n}, w')$, we obtain :

$$2w(R_{\mathcal{P}'}) \leq w(A) + w(C) \qquad [1.10]$$

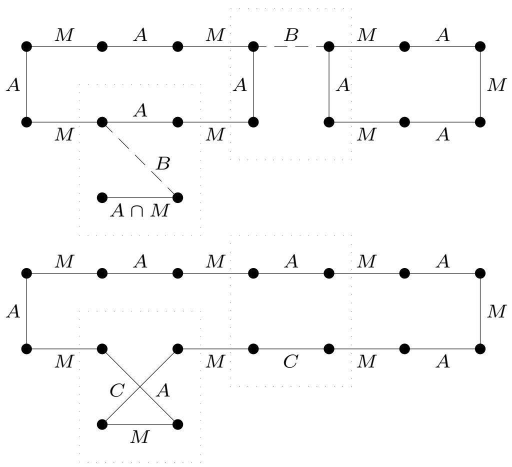


Figure 1.10. *The construction of sets A and C*

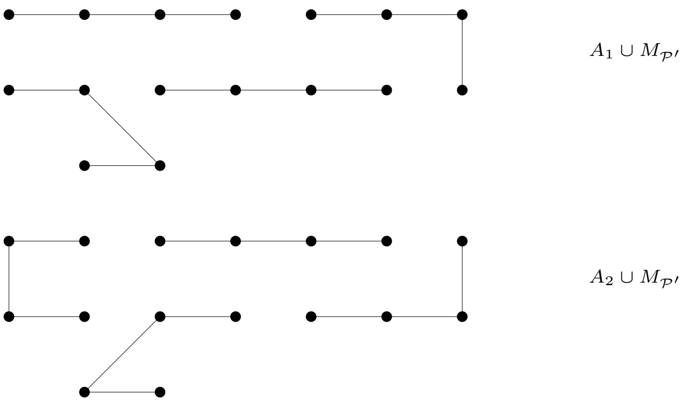


Figure 1.11. *Two possible $\mathbf{P}_4$ partitions deduced from $A \cup C \cup M_{\mathcal{P}'}$*

Moreover, item (ii) of Lemma 2 states that $B \cup D$ is a perfect matching ; since $M_{\mathcal{P}'}$ is a minimum weight perfect matching, we deduce :

$$w(M_{\mathcal{P}'}) \leq w(B) + w(D) \qquad [1.11]$$

Hence, summing up inequalities [1.10] and [1.11] (and also considering item (i) of Lemma 2), we get :

$$w(M_{\mathcal{P}'}) + 2w(R_{\mathcal{P}'}) \leq w(\mathcal{P}^*) + w(\overline{R}_{\mathcal{P}^*}) \qquad [1.12]$$

Inequality [1.12], combined with the observation that $w(\overline{R}_{\mathcal{P}^*}) \leq w(\mathcal{P}^*)$ (which is true from the assumption that I satisfies the triangle inequality), leads to the following new inequality :

$$w(M_{\mathcal{P}'}) + 2w(R_{\mathcal{P}'}) \leq 2\mathrm{opt}_{\mathrm{MINMETRIC}\mathbf{P}_4\mathrm{P}}(I) \qquad [1.13]$$

Relation [1.13] together with $w(M_{\mathcal{P}'}) \leq w(M_{\mathcal{P}^*}) \leq w(\mathcal{P}^*)$ complete the proof. Finally, the tightness is provided by the instance family $I_n = (K_{8n}, w)$ that has been described in Property 4. ■

Concerning the maximization case and using Lemma 2, one can also obtain an alternative proof of the result given in [HAS 97].

THEOREM 9.– *The solution $\mathcal{P}'$ provided by the algorithm achieves a 3/4-standard approximation for* MAX$\mathbf{P}_4$P.

PROOF.– The inequality [1.12] becomes

$$w(M_{\mathcal{P}'}) + 2w(R_{\mathcal{P}'}) \geq \mathrm{opt}_{\mathrm{MAX}\mathbf{P}_4\mathrm{P}}(I) + w(\overline{R}_{\mathcal{P}^*}) \qquad [1.14]$$

Since $M_{\mathcal{P}'}$ is a maximum weight perfect matching, the approximate value obviously satisfies $2\times w(M_{\mathcal{P}'}) \geq \mathrm{opt}_{\mathrm{MAX}\mathbf{P}_4\mathrm{P}}(I)+w(\overline{R}_{\mathcal{P}^*})$; hence, we deduce $\mathrm{apx}_{\mathrm{MAX}\mathbf{P}_4\mathrm{P}}(I) \geq \frac{3}{4}\left(\mathrm{opt}_{\mathrm{MAX}\mathbf{P}_4\mathrm{P}}(I) + w(\overline{R}_{\mathcal{P}^*})\right)$. ■

1.4.2.3. *General $\mathbf{P}_4$P within the differential framework*

When dealing with the differential ratio, MIN$\mathbf{P}_4$P, MINMETRIC$\mathbf{P}_4$P, and MAX$\mathbf{P}_4$P are equivalent to approximate, since $\mathbf{P}_k$P problems belong to the class *FGNPO*, [MON 02a]. Note that such an equivalence is more generally true for any couple of problems that only differ by an affine transformation of their objective function.

THEOREM 10.– *The solution $\mathcal{P}'$ provided by the algorithm achieves a 1/2-differential approximation for $\mathbf{P}_4$P and this ratio is tight.*

PROOF.– We consider the maximization version. First, observe that $\overline{R}_{\mathcal{P}^*}$ is an n-cardinality matching. Let M be any perfect matching of I such that $M \cup \overline{R}_{\mathcal{P}^*}$ forms a $\mathbf{P}_4$-partition, we have :

$$w(M) + w(\overline{R}_{\mathcal{P}^*}) \geq \mathrm{wor}_{\mathrm{MAX}\mathbf{P}_4\mathrm{P}}(I) \qquad [1.15]$$

Adding inequalities [1.14] and [1.15], and since $w(M_{\mathcal{P}'}) \geq w(M)$, we conclude that :

$$2\mathrm{apx}_{\mathrm{MAX}\mathbf{P}_4\mathrm{P}}(I) = 2\left(w(M_{\mathcal{P}'}) + w(R_{\mathcal{P}'})\right) \geq \mathrm{wor}_{\mathrm{MAX}\mathbf{P}_4\mathrm{P}}(I) + \mathrm{opt}_{\mathrm{MAX}\mathbf{P}_4\mathrm{P}}(I)$$

$$\Rightarrow \frac{\mathrm{apx}_{\mathrm{MAX}\mathbf{P}_4\mathrm{P}}(I) - \mathrm{wor}_{\mathrm{MAX}\mathbf{P}_4\mathrm{P}}(I)}{\mathrm{opt}_{\mathrm{MAX}\mathbf{P}_4\mathrm{P}}(I) - \mathrm{wor}_{\mathrm{MAX}\mathbf{P}_4\mathrm{P}}(I)} \geq 1/2$$

In order to establish the tightness of this ratio, we refer again to Property 4. ■

1.4.2.4. *Bi-valued metric* $\mathbf{P}_4$P *with weights 1 & 2 within the standard framework*

As it has been recently done for MINTSP in [BER 06, BL$\tilde{0}$5] and because such an analysis enables a keener comprehension of a given algorithm, we now focus on instances where any edge weight is either 1 or 2. Note that, since the $\mathbf{P}_4$-partition problem is **NP**-complete, the problems MAX$\mathbf{P}_4\mathrm{P}_{1,2}$ and MIN$\mathbf{P}_4\mathrm{P}_{1,2}$ still are **NP**-hard.

Let us first introduce some more notation. For a given instance $I = (K_{4n}, w)$ of $\mathbf{P}_4\mathrm{P}_{1,2}$ with $w(e) \in \{1, 2\}$, we denote by $M_{\mathcal{P}',i}$ (*resp.*, by $R_{\mathcal{P}',i}$) the set of edges from $M_{\mathcal{P}'}$ (*resp.*, from $R_{\mathcal{P}'}$) that are of weight i. If we aim at maximizing, then p (*resp.*, q) indicates the cardinality of $M_{\mathcal{P}',2}$ (*resp.*, of $R_{\mathcal{P}',2}$) ; otherwise, it indicates the quantity $|M_{\mathcal{P}',1}|$ (*resp.*, $|R_{\mathcal{P}',1}|$). In any case, p and q respectively count the number of "*optimum weight edges*" in the sets $M_{\mathcal{P}'}$ and $R_{\mathcal{P}'}$. With respect to the optimal solution, we define the sets $M_{\mathcal{P}^*,i}$, $R_{\mathcal{P}^*,i}$ for $i = 1, 2$ and the cardinalities p^*, q^* as the same. Wlog., we may assume that the following property always holds for $\mathcal{P}^*$:

PROPERTY 5.– *For any 3-edge path* $P \in \mathcal{P}^*$*,*
$|P \cap M_{\mathcal{P}^*,2}| \geq |P \cap R_{\mathcal{P}^*,2}|$ *if the goal is to maximize,*
$|P \cap M_{\mathcal{P}^*,1}| \geq |P \cap R_{\mathcal{P}^*,1}|$ *if the goal is to minimize.*

PROOF.– Assume that the goal is to maximize. If $|P \cap M_{\mathcal{P}^*,2}| < |P \cap R_{\mathcal{P}^*,2}|$, then $\mathcal{P}^*$ would contain a path $P = \{[x, y], [y, z], [z, t]\}$ with $w(x, y) = w(z, t) = 1$ and $w(y, z) = 2$; thus, by swapping P for $P' = \{[y, z], [z, t], [t, x]\}$ within $\mathcal{P}^*$, one could generate an alternative optimal solution. ■

LEMMA 3.– *For any instance* $I = (K_{4n}, w)$*, if* $\mathcal{P}'$ *is a feasible solution and* $\mathcal{P}^*$ *is an optimal solution, then there exists an edge set* A *that verifies :*

(i) $A \subseteq M_{\mathcal{P}^*,2} \cup R_{\mathcal{P}^*,2}$ (resp., $A \subseteq M_{\mathcal{P}^*,1} \cup R_{\mathcal{P}^*,1}$*) and* $|A| = q^*$ *if the goal is to maximize (*resp.*, to minimize) ;*

(ii) $G' = (V, M_{\mathcal{P}'} \cup A)$ *is a simple graph made of pairwise disjoint paths.*

PROOF.– We only prove the maximization case. We now consider G' the multi-graph induced by $M_{\mathcal{P}'} \cup R_{\mathcal{P}^*,2}$ (the edges from $M_{\mathcal{P}'} \cap R_{\mathcal{P}^*,2}$ appear twice). This graph consists of elementary cycles and paths : its cycles alternate edges from $M_{\mathcal{P}'}$ and $R_{\mathcal{P}^*,2}$ (in particular, the 2-edge cycles correspond to the edges from $R_{\mathcal{P}^*,2} \cap M_{\mathcal{P}'}$) ; its paths (that may be of length 1) also alternate edges from $M_{\mathcal{P}'}$ and $R_{\mathcal{P}^*,2}$, with the particularity that their end edges all belong to $M_{\mathcal{P}'}$.

Let Γ be a cycle on G' and e be an edge from $\Gamma \cap R_{\mathcal{P}^*,2}$. If $P_{\mathcal{P}^*}(e) = \{x, y, z, t\}$ denotes the path from the optimal solution that contains e, then $e = [y, z]$. The initial vertex x of the path $P_{\mathcal{P}^*}(e)$ necessarily is the endpoint of some path from G' : otherwise, the edge $[x, y]$ from $P_{\mathcal{P}^*}(e) \cap M_{\mathcal{P}^*}$ would be incident to 2 distinct edges from $R_{\mathcal{P}^*}$, which would contradict the fact that $\mathcal{P}^*$ is a $\mathbf{P}_4$ partition. The same obviously

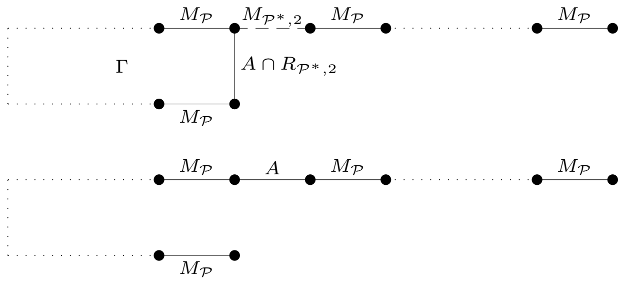


Figure 1.12. *The construction of set A*

holds for t. W.l.o.g., we may assume from Property 5 that $[x, y] \in M_{\mathcal{P}^*,2}$. In light of these remarks and in order to build an edge set A that fulfills the requirements (i) and (ii), we proceed as follows :

`Combining matchings`

1 Set $A = R_{\mathcal{P}^*,2}$; Set $G' = (V, A \cup M_{\mathcal{P}'})$ (consider the multi-graph) ;

2 While there exists a cycle Γ in G', do :

2.1 | Pick e from $\Gamma \cap R_{\mathcal{P}^*,2}$;
| Pick f from $P_{\mathcal{P}^*}(e) \cap M_{\mathcal{P}^*,2}$;
| $A \longleftarrow A \setminus \{e\} \cup \{f\}$;

2.2 $G' \longleftarrow (V, A \cup M_{\mathcal{P}'})$;

3 Output A.

By construction, the set A output by the algorithm is of cardinality q^* and contains exclusively edges of weight 2. Furthermore, each iteration of step 2 merges a cycle and a path of $A \cup M_{\mathcal{P}}$ into a path (an illustration of this merging operation is provided by Figure 1.12). Hence, the stopping criterion ensures that, at the end of this step, $G' = (V, A \cup M_{\mathcal{P}})$ is a simple graph whose connected components are elementary paths. Finally, the existence of edge f at step 2.1 directly comes from Property 5. ■

THEOREM 11.– *The solution $\mathcal{P}'$ provided by the algorithm achieves a 9/10-standard approximation for* $\mathrm{MAX}\mathbf{P}_4\mathrm{P}_{1,2}$ *and a 7/6-standard approximation for* $\mathrm{MIN}\mathbf{P}_4\mathrm{P}_{1,2}$. *These ratios are tight.*

PROOF.– Let consider A the edge subset of the optimal solution that may be deduced from the application of Lemma 3 to the approximate solution. We arbitrarily complete A by means of an edge set B so that $A \cup B \cup M_{\mathcal{P}'}$ constitutes a perfect 2-matching. As we did while proving Theorem 8, we split the edge set $A \cup B$ into two sets A_1 and A_2 in order to obtain two $\mathbf{P}_4$-partitions $M_{\mathcal{P}'} \cup A_1$ and $M_{\mathcal{P}'} \cup A_2$ of $V(K_{4n})$. As both A_1 and A_2 complete $M_{\mathcal{P}'}$ into a $\mathbf{P}_4$-partition and because $R_{\mathcal{P}'}$ is optimal, we deduce that A_i does not contain more "*optimum weight edges*" than $R_{\mathcal{P}'}$, that is : $q \geq |\{e \in A_i : w(e) = 2\}|$ if the goal is to maximize, $q \geq |\{e \in A_i : w(e) = 1\}|$ otherwise. Since $A \subseteq A_1 \cup A_2$ and $|A| = q^*$, we immediately deduce :

$$q \geq q^*/2 \tag{1.16}$$

On the other hand, by the optimality of $M_{\mathcal{P}'}$:

$$p \geq \max\{p^*, q^*\} \tag{1.17}$$

Moreover, the quantities p^* and q^* structurally verify :

$$n \geq \max\{p^*/2, q^*\} \tag{1.18}$$

Finally, we can express the value of any solution $\mathcal{P}$ as :

$$w(\mathcal{P}) = 3n + (p+q) \text{ (if goal} = \max\text{)}, 6n - (p+q) \text{ (if goal} = \min\text{)} \tag{1.19}$$

The claimed results can now be obtained from inequalities [1.16], [1.17], [1.18] and [1.19] :

$$\begin{array}{rcl}
10\mathrm{apx}_{\textsc{Max}\mathbf{P}_4\mathrm{P}_{1,2}}(I) & = & 10(3n+p+q) \\
& = & 9(3n) \;+\; 3n \;+\; 9p \;+\; p \;+\; 10q \\
& \geq & 9(3n) \;+\; 3q^* \;+\; 9p^* \;+\; q^* \;+\; 5q^* \\
& = & 9(3n+p^*+q^*) = 9\mathrm{opt}_{\textsc{Max}\mathbf{P}_4\mathrm{P}_{1,2}}(I) \\
6\mathrm{apx}_{\textsc{Min}\mathbf{P}_4\mathrm{P}_{1,2}}(I) & = & 6(6n-p-q) \\
& = & 6(6n) \;-\; 6p \;-\; 6q \\
& \leq & 6(6n) \;-\; 6p^* \;-\; 3q^* \\
& \leq & 6(6n) \;-\; 6p^* \;-\; 3q^* \;+\; (2n-p^*)+4(n-q^*) \\
& \leq & 7(6n-p^*-q^*) = 7\mathrm{opt}_{\textsc{Min}\mathbf{P}_4\mathrm{P}_{1,2}}(I)
\end{array}$$

The tightness for $\textsc{Max}\mathbf{P}_4\mathrm{P}_{1,2}$ is established in the instance $I = (K_8, w)$ depicted in Figure 1.13, where the edges of weight 2 are drawn in continuous line, and the edges of weight 1 on $\mathcal{P}^*$ and $\mathcal{P}'$ are drawn in dotted line (the other edges are not drawn). One can easily see : $\mathrm{opt}_{\textsc{Max}\mathbf{P}_4\mathrm{P}_{1,2}}(I) = 10$ and $\mathrm{apx}_{\textsc{Max}\mathbf{P}_4\mathrm{P}_{1,2}}(I) = 9$. Concerning the minimization case, the ratio is tight on the instance $J = (K_8, w)$ that verifies : $\mathrm{opt}(J) = w(\mathcal{P}^*) = 6$ and $\mathrm{apx}(J) = w(\mathcal{P}') = 7$. $J = (K_8, w)$ is depicted in Figure 1.14 (the 1-weight edges are drawn in continuous line and the 2-weight edges on $\mathcal{P}^*$ and $\mathcal{P}'$ are drawn in dotted line). ■

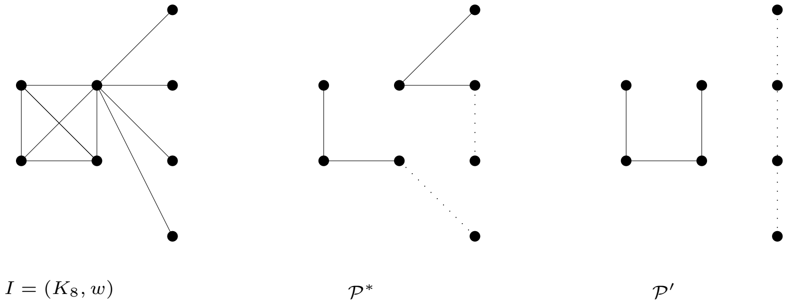


Figure 1.13. *Instance $I = (K_8, w)$ that establishes the tightness for* $\text{MAX}\mathbf{P}_4\text{P}_{1,2}$

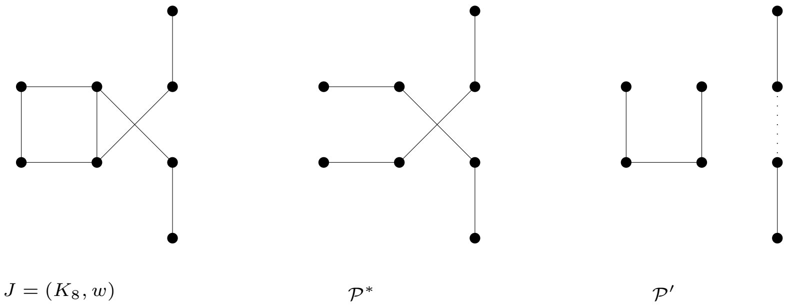


Figure 1.14. *Instance $I = (K_8, w)$ that establishes the tightness for* $\text{MIN}\mathbf{P}_4\text{P}_{1,2}$

1.4.2.5. *Bi-valued metric $\mathbf{P}_4$P with weights a and b within the differential framework*

As we have already mentioned, the differential measure is invariant under affine transformation ; now, any instance from $\text{MAX}\mathbf{P}_4\text{P}_{a,b}$ or from $\text{MIN}\mathbf{P}_4\text{P}_{a,b}$ can be mapped into an instance of $\text{MAX}\mathbf{P}_4\text{P}_{1,2}$ by the way of such a transformation. Thus, proving $\text{MAX}\mathbf{P}_4\text{P}_{1,2}$ is ε-differential approximable actually establishes that $\text{MIN}\mathbf{P}_4\text{P}_{a,b}$ and $\text{MAX}\mathbf{P}_4\text{P}_{a,b}$ are ε-differential approximable for any couple of real values $a < b$. We demonstrate here that Hassin and Rubinstein algorithm achieves a 2/3-differential approximation for $\mathbf{P}_4\text{P}_{1,2}$ and hence, for $\mathbf{P}_4\text{P}_{a,b}$, for any couple of reals $a < b$.

Let $I = (K_{4n}, w)$ be an instance of $\text{MAX}\mathbf{P}_4\text{P}_{1,2}$. We recall the notation introduced while proving Theorem 11 : $p = |M_{\mathcal{P}',2}|$, $p^* = |M_{\mathcal{P}^*,2}|$, $q = |R_{\mathcal{P}',2}|$ and $q^* = |R_{\mathcal{P}^*,2}|$. Furthermore, for $i = 1, 2$, $\mathcal{F}^i$ will refer to the set of paths from $\mathcal{P}'$ whose central edge is of weight i. Note that the paths from $\mathcal{F}^1$ may be of total weight 3, 4 or 5, whereas the paths from $\mathcal{F}^2$ may be of total weight 5 or 6 (at least one extremal edge must be of weight 2, or $M_{\mathcal{P}'}$ is not an optimum weight matching). We will denote by

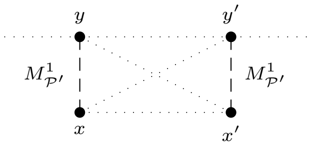


Figure 1.15. *1-weight edges on $V(M^1_{\mathcal{P}'})$*

$\mathcal{F}^2_5$ and $\mathcal{F}^2_6$ the paths from $\mathcal{F}^2$ that are of total weight 5 and 6, respectively. Finally, for $i = 1, 2$, $M^i_{\mathcal{P}'}$ will refer to the set of edges $e \in M_{\mathcal{P}'}$ such that $P_{\mathcal{P}'}(e) \in \mathcal{F}^i$ (that is, e is element of a path from $\mathcal{P}'$ whose central edge has weight i). By [1.16] and [1.17], we get :

$$\mathrm{opt}_{\mathrm{MAX}\mathbf{P}_4\mathrm{P}_{1,2}}(I) \leq \min\{3n + p + 2q, 3n + 2p\} \qquad [1.20]$$

To obtain a differential approximation, one also has to produce an efficient bound for $\mathrm{wor}_{\mathrm{MAX}\mathbf{P}_4\mathrm{P}_{1,2}}(I)$. To do so, we exploit the optimality of $M_{\mathcal{P}'}$ and $R_{\mathcal{P}'}$ in order to exhibit some edges of weight 1 that will enable us to approximate the worst solution. We first consider the vertices from $V(\mathcal{F}^1)$: they are "*easy*" to cover by means of 3-edge paths of total weight 3, since we may immediately deduce from the optimality of $R_{\mathcal{P}'}$ the following property (an illustration is provided by Figure 1.15, where dotted lines indicate edges of weight 1 and dashed lines indicate unspecified weight edges) :

PROPERTY 6.– $[x, y] \neq [x', y'] \in M^1_{\mathcal{P}'} \Rightarrow \forall (u, v) \in \{x, y\} \times \{x', y'\}\,,\ w(u, v) = 1$

We now consider the vertices from $V(\mathcal{F}^2_5)$. Let $P = \{x, y, z, t\}$ with $[x, y] \in M_{\mathcal{P}',2}$ be a path from $\mathcal{F}^2_5$, we deduce from the optimality of $M_{\mathcal{P}'}$ that $w(t, x) = 1$; hence, the 3-edge path $P' = \{y, z, t, x\}$ covers the vertices $\{x, y, z, t\}$ with a total weight 4. Let us assume that $\mathcal{F}^2_6 = \emptyset$, then we are able to build a $\mathbf{P}_4$ partition of $V(K_{4n})$ using $3n - |\mathcal{F}^2_5|$ edges of weight 1 and $|\mathcal{F}^2_5|$ edges of weight 2 (one edge of weight 2 is used for each path from $\mathcal{F}^2_5$). Hence, a worst solution costs at most $3n + q$, while the approximate solution is of total weight $3n + p + q$. Thus, using relation [1.20], we would be able to conclude that $\mathcal{P}'$ is a (2/3)-approximation. Of course, there is no reason for $\mathcal{F}^2_6 = \emptyset$; nevertheless, this discussion has brought to the fore the following fact : the difficult point of the proof lies in the partitioning of $V(\mathcal{F}^2_6)$ into "*light*" 3-edge paths. In order to deal with these vertices, we first state two more properties that are immediate from the optimality of $M_{\mathcal{P}'}$ and $R_{\mathcal{P}'}$, respectively.

PROPERTY 7.– $\left\{\begin{array}{ll} & [x, y] \in M_{\mathcal{P}',1} \textit{ and } [x', y'] \in M_{\mathcal{P}',2} \\ \Rightarrow & \min\{w(x, x'), w(y, y')\} = \min\{w(x, y'), w(y, x')\} = 1 \end{array}\right.$

PROPERTY 8.– *If* $[x, y] \neq [x', y'] \in M^1_{\mathcal{P}'}$ *and* $P_{\mathcal{P}'} = \{\alpha, \beta, \gamma, \delta\} \in \mathcal{F}^2$*, then :*
$\left\{\begin{array}{ll} & \max\{w(u, v) | (u, v) \in \{\alpha, \beta\} \times \{x, y\}\} = 2 \\ \Rightarrow & \max\{w(u, v) | (u, v) \in \{\gamma, \delta\} \times \{x', y'\}\} = 1 \end{array}\right.$

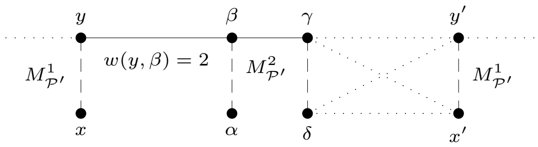


Figure 1.16. *1-weight edges that may be deduced from the optimality of $R_{\mathcal{P}'}$*

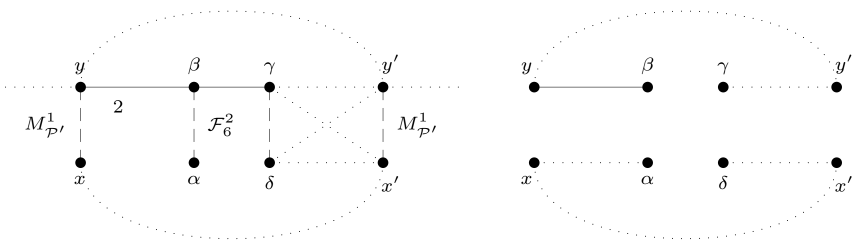


Figure 1.17. *A $\mathbf{P}_4$ partition of $(P, e_1, e_2) \in \mathcal{F}^2_6 \times (M^1_{\mathcal{P}'})^2$ of total weight at most 7*

An illustration of this latter Property is proposed in Figure 1.16, where continuous and dotted lines respectively indicate 2- and 1-weight edges, whereas dashed lines indicate unspecified weight edges. Properties 7 and 8 give the clue on how to incorporate the vertices of $\mathcal{F}^2_6$ into a packing of "*light*" $\mathbf{P}_4$. The construction of these paths is formalized in the following Property and illustrated in Figure 1.17.

PROPERTY 9.– *Given a path $P \in \mathcal{F}^2_6$ and two edges $[x, y] \neq [x', y'] \in M^1_{\mathcal{P}'}$, there exists a $\mathbf{P}_4$ partition $\mathcal{F} = \{P_1, P_2\}$ of $\big(V(P) \cup \{x, y, x', y'\}\big)$ that is of total weight at most* 8. *Furthermore, if $[x, y]$ and $[x', y']$ both belong to $M_{\mathcal{P}',1}$, then we can decrease this weight down to (at most)* 7.

PROOF.– Consider $P = \{\alpha, \beta, \gamma, \delta\} \in \mathcal{F}^2_6$ and $[x, y] \neq [x', y'] \in M^1_{\mathcal{P}'}$. We set $P_1 = \{\alpha, x, x', \delta\}$ and $P_2 = \{\beta, y, y', \gamma\}$. We know from Property 6 that $w(x, x') = w(y, y') = 1$. Thus, if every edge from $\{\alpha, \beta, \gamma, \delta\} \times \{x, x', y, y'\}$ is of weight 1, then $P_1 \cup P_2$ has a total weight 6. Conversely, if there exists a 2-weight edge that links a vertex from $\{\alpha, \beta, \gamma, \delta\}$ to a vertex from $\{x, x', y, y'\}$, we may assume that $[\beta, y]$ is such an edge; we then deduce from Property 8 that $w(\delta, x') = w(\gamma, y') = 1$ and hence, that $P_1 \cup P_2$ is of total weight at most 8. Finally, if $w(x, y) = 1$, then $w(\alpha, x) = 1$ from Property 7 and thus, $w(P_1) + w(P_2) = 7$. ■

We now are able to compute an approximate worst solution that provides an efficient upper bound for $\mathrm{wor}_{\textsc{Max}\mathbf{P}_4\mathrm{P}_{1,2}}(I)$.

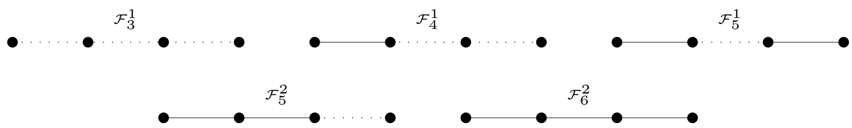


Figure 1.18. *A partition of $\mathcal{P}'$*

LEMMA 4.– *Let $I = (K_{4n}, w)$ be an instance of* MIN$\mathbf{P}_4$P$_{1,2}$ *and let $\mathcal{P}'$ be the solution provided by Hassin and Rubinstein algorithm on I. One can compute on I a solution $\mathcal{P}_*$ that verifies :*

$$p_* + q_* \leq q + (|\mathcal{F}_6^2| - \lfloor p_1^1/2 \rfloor)^+ + (|\mathcal{F}_6^2| - n + q)^+$$

where p_*, q_* and p_1^1 are defined as $p_* = |M_{\mathcal{P}_*,2}|$, $q_* = |R_{\mathcal{P}_*,2}|$ and $p_1^1 = |M_{\mathcal{P}'}^1 \cap M_{\mathcal{P}',1}|$ (and expression X^+ is equivalent to $\max\{X, 0\}$).

PROOF.– The proof is algorithmic, based on algorithm `Approximate Worst P`$_4$`P`. Note that, even though this has no impact on the rightness of the proof, the computation of $\mathcal{P}_*$ has a polynomial runtime. This means that the good properties of the approximate solution $\mathcal{P}'$ enable to really exhibit an approximate worst solution (and not only to provide an evaluation of such a solution, as it is often the case while stating differential approximation results).

In order to estimate the value of the approximate worst solution $\mathcal{P}_*$, one has to count the number $p_* + q_*$ of 2-weight edges it contains. Let p_i^1 refer to $|M_{\mathcal{P}'}^1 \cap M_{\mathcal{P}',i}|$ for $i = 1, 2$ (the cardinality p_1^1 enables the expression of the number of iterations during step 1). Steps 1, 2 and 3 respectively put into $\mathcal{P}_*$ at most one, two and three 2-weight edges per iteration. Any path from $\mathcal{F}_6^2$ is treated by one of the three steps 1, 2 and 3. If $2|\mathcal{F}_6^2| \geq p_1^1$, only $|\mathcal{F}_6^2| - \lfloor p_1^1/2 \rfloor$ paths from $\mathcal{F}_6^2$ are treated by one of the steps 2 and 3. Finally, if $|\mathcal{F}_6^2| \geq |\mathcal{F}^1|$, only $|\mathcal{F}_6^2| - |\mathcal{F}^1|$ paths from $\mathcal{F}_6^2$ are treated during step 3. Furthermore, step 4 puts at most $|\mathcal{F}_5^2|$ 2-weight edges into $\mathcal{P}_*$ (at most one per iteration), while steps 0 and 5 do not incorporate any 2-weight edges within $\mathcal{P}_*$. Thus, considering $q = |\mathcal{F}_5^2| + |\mathcal{F}_6^2|$ and $|\mathcal{F}^1| = n - q$, we obtain the announced result. ∎

Let us introduce some more notation. Analogously to $\mathcal{F}^2 = \mathcal{F}_5^2 \cup \mathcal{F}_6^2$, we define a partition of $\mathcal{F}^1$ into three subsets $\mathcal{F}_3^1$, $\mathcal{F}_4^1$ and $\mathcal{F}_5^1$ according to the path total weight. Note that, since the subsets $\mathcal{F}_j^1$ define a partition of $\mathcal{P}'$, we have $n = |\mathcal{F}_3^1| + |\mathcal{F}_4^1| + |\mathcal{F}_5^1| + |\mathcal{F}_5^2| + |\mathcal{F}_6^2|$ (see Figure 1.18 for an illustration of this partition ; the edges of weight 2 are drawn in continuous lines whereas the edges of weight 1 are drawn in dotted lines).

`Approximate Worst P`$_4$`P`

0 Set $\mathcal{P} = \mathcal{P}',\ \mathcal{P}_* = \emptyset$;

1 While $\exists \{P, e_1, e_2\} \subseteq \mathcal{P}\ s.t.\ (P, e_1, e_2) \in \mathcal{F}_6^2 \times M^1_{\mathcal{P}',1} \times M^1_{\mathcal{P}',1}$

1.1 Compute $\mathcal{F} = \{P_1, P_2\}$ on $V(P) \cup V(e_1) \cup V(e_2)$ with $w(\mathcal{F}) \leq 7$ according to Property 9 ;

1.2 $\mathcal{P} \longleftarrow \mathcal{P} \setminus \{P, e_1, e_2\}$, $\mathcal{P}_* \longleftarrow \mathcal{P}_* \cup \{P_1, P_2\}$;

2 While $\exists \{P, e_1, e_2\} \subseteq \mathcal{P}\ s.t.\ (P, e_1, e_2) \in \mathcal{F}_6^2 \times M^1_{\mathcal{P}'} \times M^1_{\mathcal{P}'}$

2.1 Compute $\mathcal{F} = \{P_1, P_2\}$ on $V(P) \cup V(e_1) \cup V(e_2)$ with $w(\mathcal{F}) \leq 8$ according to Property 9 ;

2.2 $\mathcal{P} \longleftarrow \mathcal{P} \setminus \{P, e_1, e_2\}$, $\mathcal{P}_* \longleftarrow \mathcal{P}_* \cup \{P_1, P_2\}$;

3 While $\exists P \subseteq \mathcal{P}\ s.t.\ P \in \mathcal{F}_6^2$

3.1 $\mathcal{P} \longleftarrow \mathcal{P} \setminus P,\ \mathcal{P}_* \longleftarrow \mathcal{P}_* \cup \{P\}$;

4 While $\exists P \subseteq \mathcal{P}\ s.t.\ P \in \mathcal{F}_5^2$

4.1 Compute $\mathcal{F} = \{P_1\}$ on $V(P)$ with $w(\mathcal{F}) \leq 4$;

4.2 $\mathcal{P} \longleftarrow \mathcal{P} \setminus P,\ \mathcal{P}_* \longleftarrow \mathcal{P}_* \cup \{P_1\}$;

5 While $\exists \{e_1, e_2\} \subseteq \mathcal{P}\ s.t.\ (e_1, e_2) \in M^1_{\mathcal{P}'} \times M^1_{\mathcal{P}'}$

5.1 Compute $\mathcal{F} = \{P_1\}$ on $V(e_1) \cup V(e_2)$ with $w(\mathcal{F}) = 3$;

5.2 $\mathcal{P} \longleftarrow \mathcal{P} \setminus e_1, e_2,\ \mathcal{P}_* \longleftarrow \mathcal{P}_* \cup \{P_1\}$;

6 Output $\mathcal{P}_*$.

The following Lemma states three relations between the couples of quantities (p, q), (p^*, q^*) and (p_*, q_*) that determine the value of the approximate solution, the considered optimal solution and the approximate worst solution, respectively.

LEMMA 5.–

$$p \geq q^* + (|\mathcal{F}_6^2| - \lfloor p_1^1/2 \rfloor)^+ \qquad [1.21]$$

$$2q \geq q^* + (|\mathcal{F}_6^2| + q - n)^+ \qquad [1.22]$$

$$q \geq p_* + q_* - (|\mathcal{F}_6^2| - \lfloor p_1^1/2 \rfloor)^+ - (|\mathcal{F}_6^2| + q - n)^+ \qquad [1.23]$$

PROOF.– Inequality [1.21] : Obvious if $|\mathcal{F}_6^2| \leq \lfloor p_1^1/2 \rfloor$, since $p \geq q^*$ (inequality [1.17]). Otherwise, one can write p as the sum $p = n + |\mathcal{F}_6^2| + |\mathcal{F}_5^1| - |\mathcal{F}_3^1|$. Then observe that $|\mathcal{F}_5^1| - |\mathcal{F}_3^1|$ is precisely the half of the difference between the number of 2-weight and of 1-weight edges in $M^1_{\mathcal{P}'}$: indeed, $p_2^1 = |\mathcal{F}_4^1| + 2|\mathcal{F}_5^1|$ and $p_1^1 = |\mathcal{F}_4^1| + 2|\mathcal{F}_3^1|$ and thus, $p_2^1 - p_1^1 = 2(|\mathcal{F}_5^1| - |\mathcal{F}_3^1|)$. From this latter equality, we deduce that p_1^1 and p_2^1 have the same parity, or, equivalently, that $(1/2)(p_2^1 - p_1^1) = \lfloor p_2^1/2 \rfloor - \lfloor p_1^1/2 \rfloor$.

We deduce : $p = n + |\mathcal{F}_6^2| + \lfloor p_2^1/2 \rfloor - \lfloor p_1^1/2 \rfloor \geq n + |\mathcal{F}_6^2| - \lfloor p_1^1/2 \rfloor$. Just observe that $n \geq q^*$ (inequality [1.18]) in order to conclude.

Inequality [1.22] : Obvious if $|\mathcal{F}_6^2| \leq n - q$, from inequality [1.16]. Otherwise, consider that $q \geq |\mathcal{F}_6^2|$ (by definition of q and $\mathcal{F}_6^2$) and $n \geq q^*$ (inequality [1.18]) ; therefore, $q \geq |\mathcal{F}_6^2| \geq |\mathcal{F}_6^2| + (q^* - n)$.

Inequality [1.23] : Immediate from Lemma 5. ■

THEOREM 12.– *The solution $\mathcal{P}'$ provided by the algorithm achieves a 2/3-differential approximation for $\mathbf{P}_4\mathrm{P}_{a,b}$ and this ratio is tight.*

PROOF.– By summing inequalities [1.21] to [1.23], together with $2p \geq 2p^*$, we obtain the expected result :

$$\begin{array}{rcl} 3\mathrm{apx}_{\mathrm{MAX}\mathbf{P}_4\mathrm{P}}(I) & = & 3(3n + p + q) \\ & \geq & 2(3n + p^* + q^*) + (3n + p_* + q_*) \\ & = & 2\mathrm{opt}_{\mathrm{MAX}\mathbf{P}_4\mathrm{P}_{1,2}}(I) + \mathrm{wor}_{\mathrm{MAX}\mathbf{P}_4\mathrm{P}_{1,2}}(I) \end{array}$$

The tightness is provided by the instance $I = (K_8, w)$ that is shown on Figure 1.13 ; since this instance contains some vertex v such that any edge from v is of weight 2, the result follows. ■

1.5. Conclusion

Whereas both the complexity and the approximation status of bounded-size paths packing problems in bipartite graphs with maximum degree 3 have been decided here, there remain some open questions : notably, the complexity of (INDUCED) $\mathbf{P}_k$PARTITION for $k \geq 4$ and the **APX**-hardness of MAX(INDUCED)$\mathbf{P}_k$PACKING for $k \geq 4$ in planar bipartite graphs with maximum degree 3. Those questions matter because, by drawing the precise frontier between "easy" and "hard" instances of those problems, they participate to a better understanding of what make the problems tractable or intractable. However, it also matters to obtain better approximation bounds ; in particular, concerning MAXW$\mathbf{P}_k$PACKING and MINk-PATHPARTITION : as we have already mentioned, there are no specific approximation results that exploit the specific structure of these problems. Even the results we propose here are obtained by means of quite naive algorithms ; thus, one could expect better bounds using more sophisticated algorithms. Finally, an important question concerns the approximation of $\mathbf{P}_k$P, and may be more specifically the one of MINMETRIC$\mathbf{P}_k$P, because of its relations to the minimum vehicle routing problem. We were here interested in the analysis of a given algorithm, but not really in the improvement of the approximation bounds for $\mathbf{P}_k$P. However, one could expect better and moreover, the following question remains open : does the problem admit a **PTAS** ?

1.6. Bibliographie

[ARK 98] ARKIN E., HASSIN R., « On local search for weighted packing problems », *Mathematics of Operations Research*, vol. 22, p. 640-648, 1998.

[ARK 06] ARKIN E. M., HASSIN R., LEVIN A., « Approximations for minimum and min-max vehicle routing problems », *Journal of Algorithms*, vol. 59, n˚1, p. 1-18, 2006.

[AUS 80] AUSIELLO G., D'ATRI A., PROTASI M., « Structure preserving reductions among convex optimization problems », *Journal of Computational System Sciences*, vol. 21, p. 136-153, 1980.

[AUS 99] AUSIELLO G., CRESCENZI P., GAMBOSI G., KANN V., MARCHETTI-SPACCAMELA A., PROTASI M., *Complexity and Approximation (Combinatorial Optimization Problems and Their Approximability Properties)*, Springer, Berlin, 1999.

[BEL 95] BELLARE M., ROGAWAY P., « The complexity of approximating a nonlinear program », *Mathematical Programming*, vol. 69, p. 429-441, 1995.

[BER 06] BERMAN P., KARPINSKI M., « $8/7$-Approximation Algorithm for $1,2$-TSP », *SODA'06*, Miami, Florida, United States, p. 641-648, janvier 2006.

[BL0̃5] BL`ASER M., RAM L. S., « An Improved Approximation Algorithm for TSP with Distances One and Two », *FCT'05*, L
“ubeck, Germany, p. LNCS 3623 504-515, août 2005.

[CHA 00] CHAZELLE B., « A minimum spanning tree algorithm with Inverse-Ackermann type complexity », *Journal of the ACM*, vol. 47, p. 1028-1047, 2000.

[CSA 02] CSABA B., KARPINSKI M., KRYSTA P., « Approximability of dense and sparse instances of minimum 2-connectivity, TSP and path problems », *SODA*, p. 74-83, 2002.

[DEM 96] DEMANGE M., PASCHOS V. T., « On an approximation measure founded on the links between optimization and polynomial approximation theory », *Theoretical Computer Science*, vol. 158, p. 117-141, 1996.

[DYE 86] DYER M., FRIEZE A., « Planar 3DM is NP-complete », *J. Algorithms*, vol. 7, p. 174-184, 1986.

[FRA 76] FRANK A., « Some Polynomial Time Algorithms for Certain Graphs and Hypergraphs », *Proceedings of the 5th British Combinatorial Conference, Congressus Numerantium XV, Utilitas Mathematicae*, Winnipeg, p. 211-226, 1976.

[FRE 78] FREDERICKSON G. N., HECHT M. S., KIM C. E., « Approximation algorithms for some routing problems », *SIAM J. on Computing*, vol. 7, p. 178-193, 1978.

[GAR 79] GAREY M. R., JOHNSON D. S., *Computers and intractability. A guide to the theory of NP-completeness*, CA, Freeman, 1979.

[HAS 97] HASSIN R., RUBINSTEIN S., « An Approximation Algorithm for Maximum Packing of 3-Edge Paths », *Information Processing Letters*, vol. 63, p. 63-67, 1997.

[HAS 01] HASSIN R., KHULLER S., « z-approximations », *Journal of Algorithms*, vol. 41, p. 429-442, 2001.

[HAS 06] HASSIN R., RUBINSTEIN S., « An Approximation Algorithm for Maximum Triangle Packing », *Discrete Applied Mathematics*, vol. 154, n˚6, p. 971-979, 2006.

[KAN 03] KANEKO A., « A necessary and sufficient condition for the existence of a path factor every component of which is a path of length at least two », *Journal of Combinatorial Theory, Series B*, vol. 88, p. 195-218, 2003.

[KAR 06] KARPINSKI M., « Personnal communication », 2006.

[KIR 78] KIRKPATRICK D., HELL P., « On the Completeness of a Generalized Matching Problem », *STOC'78*, p. 240-245, 1978.

[LOV 86] LOVASZ L., PLUMMER M., *Matching Theory*, North-Holland, Amsterdam, 1986.

[MON 02a] MONNOT J., « Differential approximation of **NP**-hard problems with constant size feasible solutions », *RAIRO/operations research*, vol. 36, n˚4, p. 279-297, 2002.

[MON 02b] MONNOT J., « Differential approximation results for the traveling salesman and related problems », *Information Processing Letters*, vol. 82, p. 229-235, 2002.

[MON 07a] MONNOT J., TOULOUSE S., « Approximation results for the weighted P_4 partition problem », *J of Discrete Algorithms (to appear)*, 2007.

[MON 07b] MONNOT J., TOULOUSE S., « Complexity of the Pk packing problem and related problems in bipartite graphs », *Operations Research Letters (to appear)*, 2007.

[PET 94] PETRANK E., « The Hardness of Approximation : Gap Location », *Computational Complexity*, vol. 4, p. 133-157, 1994.

[STE 00] STEINER G., « On the k-Path partition problem in cographs », *Congressus Numerantium*, vol. 147, p. 89-96, 2000.

[STE 03] STEINER G., « On the k-path partition of graphs », *Theoretical Computer Science*, vol. 290, p. 2147-2155, 2003.

[VIS 92] VISHWANATHAN S., « An Approximation Algorithm for the Asymmetric Travelling Salesman Problem with Distances One and Two », *Information Processing Letter*, vol. 44, n˚6, p. 297-302, 1992.

[WAN 94] WANG H., « Path factors of bipartite graphs », *Journal of Graph Theory*, vol. 18, p. 161-167, 1994.

[YAN 97] YAN J.-H., CHANG G., HEDETNIEMI S., HEDETNIEMI S., « On the k-path partition of graphs », *Discrete Applied Mathematics*, vol. 78, p. 227-233, 1997.

Chapitre 2

Index